

Mappel

Generated by Doxygen 1.8.11

## Contents

<b>1</b>	<b>MAPPEL</b>	<b>2</b>
<b>2</b>	<b>Install</b>	<b>5</b>
<b>3</b>	<b>OMPExcceptionCatcher</b>	<b>8</b>
<b>4</b>	<b>Namespace Index</b>	<b>9</b>
4.1	Namespace List . . . . .	9
<b>5</b>	<b>Hierarchical Index</b>	<b>10</b>
5.1	Class Hierarchy . . . . .	10
<b>6</b>	<b>Class Index</b>	<b>13</b>
6.1	Class List . . . . .	13
<b>7</b>	<b>File Index</b>	<b>16</b>
7.1	File List . . . . .	16
<b>8</b>	<b>Namespace Documentation</b>	<b>20</b>
8.1	mappel Namespace Reference . . . . .	20
8.1.1	Detailed Description . . . . .	26
8.1.2	Typedef Documentation . . . . .	26
8.1.3	Function Documentation . . . . .	29
8.1.4	Variable Documentation . . . . .	45
8.2	mappel::estimator Namespace Reference . . . . .	47
8.2.1	Class Documentation . . . . .	48
8.2.2	Enumeration Type Documentation . . . . .	50
8.2.3	Function Documentation . . . . .	50
8.3	mappel::estimator::subroutine Namespace Reference . . . . .	51
8.3.1	Detailed Description . . . . .	51

8.3.2	Function Documentation	51
8.4	mappel::mcmc Namespace Reference	54
8.4.1	Function Documentation	54
8.5	mappel::methods Namespace Reference	55
8.5.1	Detailed Description	58
8.5.2	Function Documentation	59
8.6	mappel::methods::debug Namespace Reference	67
8.6.1	Function Documentation	67
8.7	mappel::methods::likelihood Namespace Reference	68
8.7.1	Function Documentation	69
8.8	mappel::methods::likelihood::debug Namespace Reference	71
8.8.1	Function Documentation	71
8.9	mappel::methods::objective Namespace Reference	72
8.9.1	Function Documentation	74
8.10	mappel::methods::objective::debug Namespace Reference	78
8.10.1	Function Documentation	78
8.11	mappel::methods::objective::openmp Namespace Reference	80
8.11.1	Function Documentation	81
8.12	mappel::methods::openmp Namespace Reference	84
8.12.1	Function Documentation	85
8.13	omp_exception_catcher Namespace Reference	90
8.13.1	Typedef Documentation	90
8.13.2	Enumeration Type Documentation	91
8.14	omp_exception_catcher::impl_ Namespace Reference	91

<b>9</b>	<b>Class Documentation</b>	<b>91</b>
9.1	mappel::ArrayShapeError Struct Reference	91
9.1.1	Detailed Description	91
9.1.2	Constructor & Destructor Documentation	92
9.2	mappel::ArraySizeError Struct Reference	92
9.2.1	Detailed Description	92
9.2.2	Constructor & Destructor Documentation	92
9.3	mappel::estimator::CGaussHeuristicEstimator< Model > Class Template Reference	93
9.3.1	Detailed Description	94
9.3.2	Constructor & Destructor Documentation	94
9.3.3	Member Function Documentation	94
9.3.4	Member Data Documentation	103
9.4	mappel::estimator::CGaussMLE< Model > Class Template Reference	105
9.4.1	Detailed Description	106
9.4.2	Constructor & Destructor Documentation	106
9.4.3	Member Function Documentation	107
9.4.4	Member Data Documentation	115
9.5	mappel::estimator::Estimator< Model > Class Template Reference	117
9.5.1	Detailed Description	119
9.5.2	Constructor & Destructor Documentation	119
9.5.3	Member Function Documentation	119
9.5.4	Friends And Related Function Documentation	127
9.5.5	Member Data Documentation	127
9.6	mappel::Gauss1DMAP Class Reference	128
9.6.1	Detailed Description	132
9.6.2	Member Typedef Documentation	133
9.6.3	Constructor & Destructor Documentation	135
9.6.4	Member Function Documentation	135

9.6.5	Member Data Documentation . . . . .	150
9.7	mappel::Gauss1DMLE Class Reference . . . . .	156
9.7.1	Detailed Description . . . . .	160
9.7.2	Member Typedef Documentation . . . . .	161
9.7.3	Constructor & Destructor Documentation . . . . .	163
9.7.4	Member Function Documentation . . . . .	163
9.7.5	Member Data Documentation . . . . .	178
9.8	mappel::Gauss1DModel Class Reference . . . . .	184
9.8.1	Detailed Description . . . . .	188
9.8.2	Member Typedef Documentation . . . . .	188
9.8.3	Constructor & Destructor Documentation . . . . .	190
9.8.4	Member Function Documentation . . . . .	190
9.8.5	Member Data Documentation . . . . .	205
9.9	mappel::Gauss1DsMAP Class Reference . . . . .	211
9.9.1	Detailed Description . . . . .	215
9.9.2	Member Typedef Documentation . . . . .	215
9.9.3	Constructor & Destructor Documentation . . . . .	217
9.9.4	Member Function Documentation . . . . .	218
9.9.5	Member Data Documentation . . . . .	233
9.10	mappel::Gauss1DsMLE Class Reference . . . . .	239
9.10.1	Detailed Description . . . . .	243
9.10.2	Member Typedef Documentation . . . . .	244
9.10.3	Constructor & Destructor Documentation . . . . .	246
9.10.4	Member Function Documentation . . . . .	246
9.10.5	Member Data Documentation . . . . .	262
9.11	mappel::Gauss1DsModel Class Reference . . . . .	268
9.11.1	Detailed Description . . . . .	272
9.11.2	Member Typedef Documentation . . . . .	272

9.11.3	Constructor & Destructor Documentation	274
9.11.4	Member Function Documentation	274
9.11.5	Member Data Documentation	290
9.12	mappel::Gauss2DMAP Class Reference	295
9.12.1	Detailed Description	300
9.12.2	Member Typedef Documentation	300
9.12.3	Constructor & Destructor Documentation	302
9.12.4	Member Function Documentation	303
9.12.5	Member Data Documentation	318
9.13	mappel::Gauss2DMLE Class Reference	325
9.13.1	Detailed Description	330
9.13.2	Member Typedef Documentation	330
9.13.3	Constructor & Destructor Documentation	332
9.13.4	Member Function Documentation	333
9.13.5	Member Data Documentation	348
9.14	mappel::Gauss2DModel Class Reference	355
9.14.1	Detailed Description	360
9.14.2	Member Typedef Documentation	360
9.14.3	Constructor & Destructor Documentation	362
9.14.4	Member Function Documentation	362
9.14.5	Member Data Documentation	378
9.15	mappel::Gauss2DsMAP Class Reference	384
9.15.1	Detailed Description	389
9.15.2	Member Typedef Documentation	389
9.15.3	Constructor & Destructor Documentation	391
9.15.4	Member Function Documentation	392
9.15.5	Member Data Documentation	410
9.16	mappel::Gauss2DsMLE Class Reference	417

9.16.1 Detailed Description . . . . .	421
9.16.2 Member Typedef Documentation . . . . .	421
9.16.3 Constructor & Destructor Documentation . . . . .	424
9.16.4 Member Function Documentation . . . . .	424
9.16.5 Member Data Documentation . . . . .	442
9.17 mappel::Gauss2DsModel Class Reference . . . . .	449
9.17.1 Detailed Description . . . . .	454
9.17.2 Member Typedef Documentation . . . . .	454
9.17.3 Constructor & Destructor Documentation . . . . .	456
9.17.4 Member Function Documentation . . . . .	456
9.17.5 Member Data Documentation . . . . .	473
9.18 mappel::Gauss2DsxyMAP Class Reference . . . . .	480
9.18.1 Detailed Description . . . . .	484
9.18.2 Member Typedef Documentation . . . . .	484
9.18.3 Constructor & Destructor Documentation . . . . .	486
9.18.4 Member Function Documentation . . . . .	486
9.18.5 Member Data Documentation . . . . .	501
9.19 mappel::Gauss2DsxyModel Class Reference . . . . .	507
9.19.1 Detailed Description . . . . .	511
9.19.2 Member Typedef Documentation . . . . .	511
9.19.3 Constructor & Destructor Documentation . . . . .	513
9.19.4 Member Function Documentation . . . . .	513
9.19.5 Member Data Documentation . . . . .	525
9.20 mappel::estimator::HeuristicEstimator< Model > Class Template Reference . . . . .	530
9.20.1 Detailed Description . . . . .	532
9.20.2 Constructor & Destructor Documentation . . . . .	532
9.20.3 Member Function Documentation . . . . .	532
9.20.4 Member Data Documentation . . . . .	541

9.21	mappel::ImageFormat1DBase Class Reference	543
9.21.1	Detailed Description	544
9.21.2	Member Typedef Documentation	545
9.21.3	Constructor & Destructor Documentation	546
9.21.4	Member Function Documentation	546
9.21.5	Member Data Documentation	548
9.22	mappel::ImageFormat2DBase Class Reference	549
9.22.1	Detailed Description	551
9.22.2	Member Typedef Documentation	551
9.22.3	Constructor & Destructor Documentation	552
9.22.4	Member Function Documentation	553
9.22.5	Member Data Documentation	555
9.23	mappel::estimator::IterativeMaximizer< Model > Class Template Reference	556
9.23.1	Detailed Description	559
9.23.2	Constructor & Destructor Documentation	559
9.23.3	Member Function Documentation	559
9.23.4	Member Data Documentation	572
9.24	mappel::LogicalError Struct Reference	577
9.24.1	Detailed Description	577
9.24.2	Constructor & Destructor Documentation	577
9.25	mappel::MAPEstimator Class Reference	577
9.25.1	Detailed Description	580
9.25.2	Member Typedef Documentation	580
9.25.3	Constructor & Destructor Documentation	580
9.25.4	Member Function Documentation	580
9.25.5	Member Data Documentation	589
9.26	mappel::estimator::IterativeMaximizer< Model >::MaximizerData Class Reference	592
9.26.1	Detailed Description	593



9.26.2	Constructor & Destructor Documentation	594
9.26.3	Member Function Documentation	594
9.26.4	Member Data Documentation	598
9.27	mappel::MCMCAdaptor1D Class Reference	601
9.27.1	Detailed Description	604
9.27.2	Member Typedef Documentation	604
9.27.3	Constructor & Destructor Documentation	604
9.27.4	Member Function Documentation	605
9.27.5	Member Data Documentation	615
9.28	mappel::MCMCAdaptor1Ds Class Reference	619
9.28.1	Detailed Description	622
9.28.2	Member Typedef Documentation	622
9.28.3	Constructor & Destructor Documentation	622
9.28.4	Member Function Documentation	623
9.28.5	Member Data Documentation	633
9.29	mappel::MCMCAdaptor2D Class Reference	637
9.29.1	Detailed Description	640
9.29.2	Member Typedef Documentation	640
9.29.3	Constructor & Destructor Documentation	640
9.29.4	Member Function Documentation	641
9.29.5	Member Data Documentation	651
9.30	mappel::MCMCAdaptor2Ds Class Reference	655
9.30.1	Detailed Description	658
9.30.2	Member Typedef Documentation	658
9.30.3	Constructor & Destructor Documentation	659
9.30.4	Member Function Documentation	659
9.30.5	Member Data Documentation	669
9.31	mappel::MCMCAdaptorBase Class Reference	674

9.31.1	Detailed Description	675
9.31.2	Constructor & Destructor Documentation	675
9.31.3	Member Function Documentation	675
9.31.4	Member Data Documentation	676
9.32	mappel::mcmc::MCMCData Struct Reference	677
9.32.1	Detailed Description	678
9.32.2	Member Function Documentation	678
9.32.3	Member Data Documentation	678
9.33	mappel::mcmc::MCMCDataStack Struct Reference	679
9.33.1	Detailed Description	680
9.33.2	Member Function Documentation	680
9.33.3	Member Data Documentation	680
9.34	mappel::mcmc::MCMCDebugData Struct Reference	682
9.34.1	Detailed Description	682
9.34.2	Member Function Documentation	682
9.34.3	Member Data Documentation	682
9.35	mappel::MLEstimator Class Reference	683
9.35.1	Detailed Description	686
9.35.2	Member Typedef Documentation	686
9.35.3	Constructor & Destructor Documentation	686
9.35.4	Member Function Documentation	686
9.35.5	Member Data Documentation	695
9.36	mappel::ModelBoundsError Struct Reference	698
9.36.1	Detailed Description	698
9.36.2	Constructor & Destructor Documentation	698
9.37	mappel::estimator::NewtonDiagonalMaximizer< Model > Class Template Reference	699
9.37.1	Detailed Description	701
9.37.2	Member Typedef Documentation	701

9.37.3	Constructor & Destructor Documentation	702
9.37.4	Member Function Documentation	702
9.37.5	Member Data Documentation	716
9.38	mappel::estimator::NewtonMaximizer< Model > Class Template Reference	720
9.38.1	Detailed Description	723
9.38.2	Member Typedef Documentation	723
9.38.3	Constructor & Destructor Documentation	723
9.38.4	Member Function Documentation	724
9.38.5	Member Data Documentation	737
9.39	mappel::NotImplementedError Struct Reference	741
9.39.1	Detailed Description	741
9.39.2	Constructor & Destructor Documentation	742
9.40	mappel::NumericalError Struct Reference	742
9.40.1	Detailed Description	742
9.40.2	Constructor & Destructor Documentation	742
9.41	omp_exception_catcher::impl_::OMPExcptionCatcher< _dummy > Class Template Reference	743
9.41.1	Detailed Description	743
9.41.2	Constructor & Destructor Documentation	743
9.41.3	Member Function Documentation	744
9.42	mappel::ParameterValueError Struct Reference	745
9.42.1	Detailed Description	745
9.42.2	Constructor & Destructor Documentation	745
9.43	mappel::PointEmitterModel Class Reference	746
9.43.1	Detailed Description	748
9.43.2	Member Typedef Documentation	749
9.43.3	Constructor & Destructor Documentation	749
9.43.4	Member Function Documentation	749
9.43.5	Member Data Documentation	758

9.44	PoissonGaussianNoise2DObjective< ModelBase > Class Template Reference	. 760
9.44.1	Detailed Description	. 761
9.44.2	Member Typedef Documentation	. 761
9.44.3	Constructor & Destructor Documentation	. 762
9.44.4	Member Data Documentation	. 762
9.45	mappell::PoissonNoise1DObjective Class Reference	. 763
9.45.1	Detailed Description	. 764
9.45.2	Member Typedef Documentation	. 764
9.45.3	Constructor & Destructor Documentation	. 766
9.45.4	Member Function Documentation	. 766
9.45.5	Member Data Documentation	. 769
9.46	mappell::PoissonNoise2DObjective Class Reference	. 770
9.46.1	Detailed Description	. 771
9.46.2	Member Typedef Documentation	. 771
9.46.3	Constructor & Destructor Documentation	. 773
9.46.4	Member Function Documentation	. 773
9.46.5	Member Data Documentation	. 776
9.47	mappell::estimator::ProfileBoundsData Struct Reference	. 777
9.47.1	Detailed Description	. 778
9.47.2	Member Function Documentation	. 778
9.47.3	Member Data Documentation	. 778
9.48	mappell::estimator::ProfileBoundsDataStack Struct Reference	. 780
9.48.1	Detailed Description	. 781
9.48.2	Member Function Documentation	. 781
9.48.3	Member Data Documentation	. 782
9.49	mappell::estimator::QuasiNewtonMaximizer< Model > Class Template Reference	. 784
9.49.1	Detailed Description	. 786
9.49.2	Member Typedef Documentation	. 786

9.49.3	Constructor & Destructor Documentation . . . . .	787
9.49.4	Member Function Documentation . . . . .	787
9.49.5	Member Data Documentation . . . . .	801
9.50	mappel::estimator::SimulatedAnnealingMaximizer< Model > Class Template Reference . . . . .	805
9.50.1	Detailed Description . . . . .	807
9.50.2	Constructor & Destructor Documentation . . . . .	807
9.50.3	Member Function Documentation . . . . .	807
9.50.4	Member Data Documentation . . . . .	815
9.51	mappel::Gauss2DsxyModel::Stencil Class Reference . . . . .	817
9.51.1	Detailed Description . . . . .	818
9.51.2	Member Typedef Documentation . . . . .	818
9.51.3	Constructor & Destructor Documentation . . . . .	819
9.51.4	Member Function Documentation . . . . .	819
9.51.5	Friends And Related Function Documentation . . . . .	820
9.51.6	Member Data Documentation . . . . .	820
9.52	mappel::Gauss1DsModel::Stencil Class Reference . . . . .	822
9.52.1	Detailed Description . . . . .	823
9.52.2	Member Typedef Documentation . . . . .	823
9.52.3	Constructor & Destructor Documentation . . . . .	823
9.52.4	Member Function Documentation . . . . .	824
9.52.5	Friends And Related Function Documentation . . . . .	825
9.52.6	Member Data Documentation . . . . .	825
9.53	mappel::Gauss2DModel::Stencil Class Reference . . . . .	826
9.53.1	Detailed Description . . . . .	827
9.53.2	Member Typedef Documentation . . . . .	827
9.53.3	Constructor & Destructor Documentation . . . . .	827
9.53.4	Member Function Documentation . . . . .	828
9.53.5	Friends And Related Function Documentation . . . . .	829

9.53.6	Member Data Documentation . . . . .	829
9.54	mappel::Gauss2DsModel::Stencil Class Reference . . . . .	831
9.54.1	Detailed Description . . . . .	832
9.54.2	Member Typedef Documentation . . . . .	832
9.54.3	Constructor & Destructor Documentation . . . . .	832
9.54.4	Member Function Documentation . . . . .	832
9.54.5	Friends And Related Function Documentation . . . . .	834
9.54.6	Member Data Documentation . . . . .	834
9.55	mappel::Gauss1DModel::Stencil Class Reference . . . . .	836
9.55.1	Detailed Description . . . . .	837
9.55.2	Member Typedef Documentation . . . . .	837
9.55.3	Constructor & Destructor Documentation . . . . .	838
9.55.4	Member Function Documentation . . . . .	838
9.55.5	Friends And Related Function Documentation . . . . .	839
9.55.6	Member Data Documentation . . . . .	839
9.56	mappel::estimator::ThreadedEstimator< Model > Class Template Reference . . . . .	840
9.56.1	Detailed Description . . . . .	842
9.56.2	Constructor & Destructor Documentation . . . . .	842
9.56.3	Member Function Documentation . . . . .	842
9.56.4	Member Data Documentation . . . . .	851
9.57	mappel::estimator::TrustRegionMaximizer< Model > Class Template Reference . . . . .	853
9.57.1	Detailed Description . . . . .	856
9.57.2	Member Typedef Documentation . . . . .	856
9.57.3	Constructor & Destructor Documentation . . . . .	857
9.57.4	Member Function Documentation . . . . .	857
9.57.5	Member Data Documentation . . . . .	871

<b>10 File Documentation</b>	<b>876</b>
10.1 display.cpp File Reference . . . . .	876
10.1.1 Detailed Description . . . . .	877
10.2 display.h File Reference . . . . .	877
10.2.1 Detailed Description . . . . .	878
10.3 estimator.cpp File Reference . . . . .	878
10.3.1 Detailed Description . . . . .	879
10.4 estimator.h File Reference . . . . .	879
10.4.1 Detailed Description . . . . .	880
10.4.2 Class Documentation . . . . .	880
10.4.3 Typedef Documentation . . . . .	882
10.5 estimator_helpers.h File Reference . . . . .	882
10.5.1 Detailed Description . . . . .	883
10.6 estimator_impl.h File Reference . . . . .	883
10.6.1 Detailed Description . . . . .	884
10.7 estimator_statics.cpp File Reference . . . . .	884
10.7.1 Detailed Description . . . . .	884
10.8 Gauss1DMAP.cpp File Reference . . . . .	884
10.8.1 Detailed Description . . . . .	885
10.9 Gauss1DMAP.h File Reference . . . . .	885
10.9.1 Detailed Description . . . . .	885
10.10Gauss1DMLE.cpp File Reference . . . . .	886
10.10.1 Detailed Description . . . . .	886
10.11Gauss1DMLE.h File Reference . . . . .	886
10.11.1 Detailed Description . . . . .	887
10.12Gauss1DModel.cpp File Reference . . . . .	887
10.12.1 Detailed Description . . . . .	887
10.13Gauss1DModel.h File Reference . . . . .	887

10.13.1 Detailed Description . . . . .	888
10.14Gauss1DsMAP.cpp File Reference . . . . .	888
10.14.1 Detailed Description . . . . .	888
10.15Gauss1DsMAP.h File Reference . . . . .	889
10.15.1 Detailed Description . . . . .	889
10.16Gauss1DsMLE.cpp File Reference . . . . .	889
10.16.1 Detailed Description . . . . .	890
10.17Gauss1DsMLE.h File Reference . . . . .	890
10.17.1 Detailed Description . . . . .	890
10.18Gauss1DsModel.cpp File Reference . . . . .	891
10.18.1 Detailed Description . . . . .	891
10.19Gauss1DsModel.h File Reference . . . . .	891
10.19.1 Detailed Description . . . . .	892
10.20Gauss2DMAP.cpp File Reference . . . . .	892
10.20.1 Detailed Description . . . . .	892
10.21Gauss2DMAP.h File Reference . . . . .	892
10.21.1 Detailed Description . . . . .	893
10.22Gauss2DMLE.cpp File Reference . . . . .	893
10.22.1 Detailed Description . . . . .	893
10.23Gauss2DMLE.h File Reference . . . . .	894
10.23.1 Detailed Description . . . . .	894
10.24Gauss2DModel.cpp File Reference . . . . .	894
10.24.1 Detailed Description . . . . .	895
10.25Gauss2DModel.h File Reference . . . . .	895
10.25.1 Detailed Description . . . . .	896
10.26Gauss2DsMAP.cpp File Reference . . . . .	896
10.26.1 Detailed Description . . . . .	896
10.27Gauss2DsMAP.h File Reference . . . . .	896



10.27.1 Detailed Description . . . . .	897
10.28Gauss2DsMLE.cpp File Reference . . . . .	897
10.28.1 Detailed Description . . . . .	897
10.29Gauss2DsMLE.h File Reference . . . . .	898
10.29.1 Detailed Description . . . . .	898
10.30Gauss2DsModel.cpp File Reference . . . . .	898
10.30.1 Detailed Description . . . . .	899
10.31Gauss2DsModel.h File Reference . . . . .	899
10.31.1 Detailed Description . . . . .	900
10.32Gauss2DsxyMAP.h File Reference . . . . .	900
10.32.1 Detailed Description . . . . .	901
10.33Gauss2DsxyModel.h File Reference . . . . .	901
10.33.1 Detailed Description . . . . .	902
10.34ImageFormat1DBase.cpp File Reference . . . . .	902
10.34.1 Detailed Description . . . . .	902
10.35ImageFormat1DBase.h File Reference . . . . .	902
10.35.1 Detailed Description . . . . .	903
10.36ImageFormat2DBase.cpp File Reference . . . . .	903
10.36.1 Detailed Description . . . . .	904
10.37ImageFormat2DBase.h File Reference . . . . .	904
10.37.1 Detailed Description . . . . .	904
10.38Install.md File Reference . . . . .	905
10.39MAPEstimator.h File Reference . . . . .	905
10.39.1 Detailed Description . . . . .	906
10.40mcmc.cpp File Reference . . . . .	906
10.40.1 Detailed Description . . . . .	906
10.41mcmc.h File Reference . . . . .	907
10.41.1 Detailed Description . . . . .	907

10.42mcmc_data.h File Reference . . . . .	908
10.42.1 Detailed Description . . . . .	908
10.43MCMCAdaptor1D.cpp File Reference . . . . .	908
10.43.1 Detailed Description . . . . .	909
10.44MCMCAdaptor1D.h File Reference . . . . .	909
10.44.1 Detailed Description . . . . .	909
10.45MCMCAdaptor1Ds.cpp File Reference . . . . .	909
10.45.1 Detailed Description . . . . .	910
10.46MCMCAdaptor1Ds.h File Reference . . . . .	910
10.46.1 Detailed Description . . . . .	910
10.47MCMCAdaptor2D.cpp File Reference . . . . .	911
10.47.1 Detailed Description . . . . .	911
10.48MCMCAdaptor2D.h File Reference . . . . .	911
10.48.1 Detailed Description . . . . .	911
10.49MCMCAdaptor2Ds.cpp File Reference . . . . .	912
10.49.1 Detailed Description . . . . .	912
10.50MCMCAdaptor2Ds.h File Reference . . . . .	912
10.50.1 Detailed Description . . . . .	912
10.51MCMCAdaptorBase.cpp File Reference . . . . .	913
10.51.1 Detailed Description . . . . .	913
10.52MCMCAdaptorBase.h File Reference . . . . .	913
10.52.1 Detailed Description . . . . .	913
10.53MLEstimator.h File Reference . . . . .	914
10.53.1 Detailed Description . . . . .	915
10.54model_methods.h File Reference . . . . .	915
10.55model_methods_impl.h File Reference . . . . .	918
10.56numerical.cpp File Reference . . . . .	921
10.56.1 Detailed Description . . . . .	922

10.57numerical.h File Reference . . . . .	922
10.57.1 Detailed Description . . . . .	923
10.58OMPExcptionCatcher.h File Reference . . . . .	923
10.58.1 Detailed Description . . . . .	924
10.59openmp_methods.h File Reference . . . . .	925
10.59.1 Detailed Description . . . . .	927
10.60PointEmitterModel.cpp File Reference . . . . .	927
10.60.1 Detailed Description . . . . .	928
10.61PointEmitterModel.h File Reference . . . . .	928
10.61.1 Detailed Description . . . . .	929
10.62PoissonGaussianNoise2DObjective.cpp File Reference . . . . .	929
10.63PoissonGaussianNoise2DObjective.h File Reference . . . . .	929
10.63.1 Detailed Description . . . . .	930
10.63.2 Function Documentation . . . . .	931
10.64PoissonNoise1DObjective.cpp File Reference . . . . .	932
10.64.1 Detailed Description . . . . .	933
10.65PoissonNoise1DObjective.h File Reference . . . . .	933
10.65.1 Detailed Description . . . . .	934
10.66PoissonNoise2DObjective.cpp File Reference . . . . .	935
10.66.1 Detailed Description . . . . .	935
10.67PoissonNoise2DObjective.h File Reference . . . . .	935
10.67.1 Detailed Description . . . . .	937
10.68README.md File Reference . . . . .	937
10.69README.md File Reference . . . . .	937
10.70rng.cpp File Reference . . . . .	937
10.70.1 Detailed Description . . . . .	937
10.71rng.h File Reference . . . . .	938
10.71.1 Detailed Description . . . . .	938
10.72stencil.cpp File Reference . . . . .	939
10.72.1 Detailed Description . . . . .	939
10.73stencil.h File Reference . . . . .	940
10.73.1 Detailed Description . . . . .	941
10.74util.cpp File Reference . . . . .	941
10.75util.h File Reference . . . . .	942
10.75.1 Detailed Description . . . . .	944

## 1 MAPPEL

Mappel is an object-oriented image processing library for high-performance [super-resolution localization](#) of Gaussian point emitters in [fluorescence microscopy](#) applications.

- Mappel uses CMake and builds cross-platform for Linux and Windows 64-bit.
- Mappel provides object-oriented interfaces for C++, Python, and Matlab.
- Mappel uses OpenMP to parallelize operations over vectors of images or parameters
- Mappel is free-as-in-beer and free-as-in-speech! ([GPL-v3](LICENSE))

### Documentation

The Mappel Doxygen documentation can be build with the `OPT_DOC` CMake option and is also available on online:

- [Mappel HTML Manual](#)
- [Mappel PDF Manual](#)
- [Mappel github repository](#)

### Background

Point emitter localization is a process of precisely estimating the sub-pixel location of a single point source emitters (molecules/proteins) at effective resolutions 10-50 times smaller than the fundamental diffraction limit for optical microscopes. Operationally, this is the process of going from blurry, noisy, pixelated images to a sub-pixel estimate of true emitter position as well as the uncertainty in that estimate. Figure 1 shows the point emitter localization process with realistic physical values for a typical super-resolution fluorescence microscope configuration.

**Figure 1:** Effective fitting resolution in typical applications

### Applications

- Stochastic super-resolution reconstruction with [PALM](#) and [dSTORM](#) florescence microscopy techniques.
- [Single particle tracking \(SPT\)](#)
  - The [Robust Particle Tracking \(RPT\)](#) library uses Mappel for the localization phase of tracking.
- [Nano-structure optical measurements](#) and alignment.
- Accurate estimation of fluorophore emitter intensity over time.

## Performance

Emitter localization applications, especially SPT and super-resolution imaging, can require millions of emitter estimations per dataset. This demand is only increasing with the drive towards larger EMCCD and SCMOS sensors and longer experiments at higher frame-rates. Speed becomes even more crucial for these applications when batch processing dozens of large data files.

- Mappel runs all image oriented computations in parallel using OpenMP making full use the system hardware concurrency.
- Mappel is fast. It can easily localize  $10^4$  emitters/sec/core on modern consumer hardware
- Small and medium-sized datasets using Mappel can work well on laptops allowing interactive Matlab applications like [RPT](#) to be used from nearly any machine.

## Installation

Mappel uses the [CMake](#) build system, and is designed to be cross-compiled from linux to other platforms, primarily Win64, although future OSX support is planned.

## Dependencies

Several standard numerical packages are required to build Mappel. Most distributions should have development versions of these packages which provide the include files and other necessary development files for the packages.

- [Armadillo](#) - A high-performance array library for C++.
- [Boost](#)
- BLAS
  - Requires support for 64-bit integers.
  - [Netlib BLAS Reference](#)
- LAPACK
  - Requires support for 64-bit integers.
  - [Netlib LAPACK Reference](#)

Note the `OPT_BLAS_INT64` CMake option controls whether Armadillo uses BLAS and LAPACK libraries that use 64-bit integer indexing. Matlab uses 64-bit by default, so linking Mappel to Matlab MEX libraries requires this option enabled. Many linux systems only provide 32-bit integer versions of BLAS and Lapack, and the option can be disabled if Matlab support is not a concern and 64-bit support is difficult to provide.

## External Projects

These packages are specialized CMake projects. If they are not currently installed on the development machines we use the `AddExternalDependency.cmake` which will automatically download, configure, build and install to the `CMAKE_INSTALL_PREFIX`, enabling their use through the normal CMake `find_package()` system.

- `BacktraceException` - A library to provide debugging output on exception calls. Important for Matlab debugging.
- `ParallelRngManager` - A simple manager for easily deploying a set of RNG parallelized over a set number of threads, using the TRNG parallel RNG library.
- `PriorHessian` - The PriorHessian library allows fast computation of log-likelihood and derivatives for composite priors.

## Model classes

Mappel provides model objects that correspond to different fitting-modes (psf-models). Mappel's core is a C++ library `libmappel.so` that uses OpenMP to automatically parallelize localizations over multiple images. Mappel also provides detailed object-oriented interfaces for Python and Matlab, using the same concept of a Model class to represent each class of psf fitting models.

## Computations available

- `llh` - log-likelihood (log of pdf)
- `rllh` - relative log-likelihood (log of pdf without constant terms)
- `grad` - derivative of log-likelihood (or equivalently of relative-llh)
- `grad2` - 2nd-derivative of log-likelihood
- `hessian` - hessian of log-likelihood

## Design Notes

### Static Polymorphism

The Mappel library is designed using static polymorphism (templates), and as such avoids virtual functions for small-grained tasks, and instead uses templates, which allow many small functions to be inlined. This aggressive inlining by the compiler produces log-likelihood, gradient, and hessian functions that are nearly as fast as hand-coded functions.

## License

### LICENSE

- Copyright: 2013-2019
- Author: Mark J. Olah
- Email: (`mjo@cs.unm.edu`)
- LICENSE: GPL-v3 See `LICENSE` file.

## 2 Install

Currently building has only been tested on linux hosts.

Mappel has been tested with modern GCC-7.2.0 as well as earlier GCC's to 4.9.4 (which is still required for Matlab integration). In order to maintain compatibility with gcc-4.9.4 and still use modern C++14 features we use the `-std=c++1y` GCC standard when compiling.

### Dependencies

Several standard numerical packages are required to build Mappel. Most distributions should have development versions of these packages which provide the include files and other necessary development files for the packages.

- [Armadillo](#)
- [Boost](#)
- [TRNG](#)
- BLAS
  - Requires support for 64-bit integers.
  - [Netlib BLAS Reference](#)
- LAPACK
  - Requires support for 64-bit integers.
  - [Netlib LAPACK Reference](#)

### Gentoo

Add to your `package.keywords`

```
1 sci-libs/lapack-reference int64
2 sci-libs/blas-reference int64
```

```
1 emerge -av armadillo boost lapack-reference blas-reference
```

For TRNG there is not a gentoo ebuild in the tree, we provide one at the [OlahGentooScienceOverlay](#).

```
1 layman -o https://github.com/markjolah/OlahScienceGentooOverlay/blob/master/layman.xml -f -a olah-science
2 emerge -av trng
```

## External Projects

Mappel also depends on several small Github projects which for now are maintained in separate repositories.

- `BacktraceException` - A library to provide debugging output on exception calls. Important for Matlab debugging.
- `ParallelRngManager` - A simple manager for easily deploying a set of RNG parallelized over a set number of threads, using the TRNG parallel RNG library.
- `PriorHessian` - The PriorHessian library allows fast computation of log-likelihood and derivatives for composite priors.

If these libraries do not exist on the build system or at `CMAKE_INSTALL_PREFIX`, they are automatically downloaded, configured and installed as external dependencies during the CMake configure phase.

Normally the Github current versions of the external dependencies are used. To use the HEAD revision of a local git repository, the following Environment variables can be set:

- `BacktraceExceptionURL` - Local directory or git URL for the BacktraceException library [optional] Default to use the HEAD version from Github
- `ParallelRngManagerURL` - Local directory or git URL for the ParallelRngManager library
- `PriorHessianURL` - Local directory or git URL for the PriorHessian library

## Build process

### Linux

```
1 git clone https://github.com/markjolah/Mappel.git
2 cd Mappel
3 ./build.sh
```

On successful build the Mappel libraries, binaries, includes, and CMake modules are all installed to the `_install` dir.

### Debugging

CMake variable `CMAKE_BUILD_TYPE=Debug` will configure the debug build and all libraries and executables will have a `.debug` suffix.

A convenience script exists to only build the debug versions of the libraries in the local `_build` and `_install` directories

```
1 ./build.debug.sh
```

### Tips:

- Try running `VERBOSE=1 make` inside the `_build/Debug` directory to debug the build/link command lines generated by CMAKE.



### Python support

Matlab support is enabled by setting the CMake option `-DOPT_PYTHON=1`. At the moment only python 3 is supported.

Several CMake variable can control for which python version modules are built.

- `MAPPEL_PYTHON_VERSIONS` - List of python X.Y versions seperated by ";" to build modules for (e.g., "3.4;3.5;3.6")
- `MAPPEL_PYTHON_EXECUTABLE` - Name or full path to python executable on the system for which to build (e.g., `python3`).

Mappel uses `pybind11` to compile modules for each Mappel Model class.

### Python development workflow

The Mappel python package environment is created in the build tree at `build_dir/python`, as part of the CMake build process. There is a standard `setuptools` `setup.py` that can be used to build binary distributions and also to install to the local system.

The CMake install process will automatically install the python `.egg` using `setup.py` under the `CMAKE_INSTALL_PREFIX` directory.

In order to be able to develop the code at the root `mappel/python/` git repository while running and testing the mappel package without having to make `install` on every small change to python code, we use the `developer mode` install option provided by `setuptools`. In fact, we have made it even easier to use, by making an alias `localdevelop`

```
1 $ cd _build/Debug/python
2 $ python setup.py localdevelop
3 $ python -m mappel
```

### Matlab support

Matlab support is enabled by setting the CMake option `MATLAB=on`. This brings in an additional external dependency,

- `MexIFace` - A cross-platform Matlab/C++ class-based interface wrapper for generating `.mex` files.

The following environment variables control the Matlab build process

- `MexIFaceURL`: Local directory or git URL for the Mexiface library (Matlab Support). [optional] Default to use the HEAD version from Github
- `MATLAB_LIBS_ROOT`: [Optional] Local path to find Matlab core shared libraries to link against (overrides default search paths). Must contain subdirectory structure (`$MATLAB_ARCH`) `//{bin,extern}`. `MATLAB_ARCH` is `[glnxa64, maci64, win64]`.
- `MATLAB_ROOT_GLNXA64`: Necessary for Matlab. Location of the Matlab `glnxa64` version to link against.

### Cross-building to Win64

The following Environment variables control the Win64 cross-build environment necessary to compile win64 binaries

- `MXE_ROOT` Local directory root of the MXE Win64 cross environment. Necessary For Win64 cross-compiling only.
- `MATLAB_ROOT_WIN64` Necessary for Matlab on Win64 cross build. Location of the Matlab win64 version to link against.

### Cross-building to OSX

- `OSXCROSS_ROOT`: Local directory root of the OSXCross OSX 64-bit cross environment. Necessary for OSX cross-compiling only.
- `MATLAB_ROOT_MACI64`: Necessary for Matlab on OSX cross build. Location of the Matlab maci64 version to link against.

## 3 OMPEXceptionCatcher

A lightweight class for managing C++ exception handling strategies in OpenMP code.

### Motivation

OpenMP code must catch any exceptions that may have been thrown before exiting the OpenMP block. This class acts as lightweight wrapper that allows an arbitrary function or lambda expression to be run safely and efficiently in OMP even if it might throw exceptions. We employ one of 4 possible strategies as determined By the `OMPEXceptionCatcher::Strategies` enum.

### Exception Catching Strategy's

- `OMPEXceptionCatcher::Strategies::DoNotTry` – Don't even try, this is a null op to completely disable this class's effect.
- `OMPEXceptionCatcher::Strategies::Continue` – Catch exceptions and keep going
- `OMPEXceptionCatcher::Strategies::Abort` – Catch exceptions and abort
- `OMPEXceptionCatcher::Strategies::RethrowFirst` – Re-throws first exception thrown by any thread

### Including OMPEXceptionCatcher in your OpenMP project

Since OMPEXceptionCatcher is header-only, the easiest way to use it is via the [git subrepo](#) plugin. Unlike the traditional `git submodule` command, `git subrepo` is transparent to other users of your repository, and solves many of the irksome issues prevalent with the submodule approach. Follow the [git subrepo install guide](#) to install on your development machine.

Then to add OMPEXceptionCatcher,

```
1 > cd $MY_REPOS
2 > git subrepo pull https://github.com/markjolah/OMPEXceptionCatcher include/where/ever/OMPEXceptionCatcher
```

### ## Example useage:

```
#include <OMPEXceptionCatcher/OMPEXceptionCatcher.h>
OMPEXceptionCatcher catcher(OMPEXceptionCatcher<>::Strategies::Continue);
#pragma omp parallel for
for(int n=0; n < N; n++)
    catcher.run([&]{ my_output(n)=do_my_calculations(args(n)); })
catcher.rethrow(); //Required only if you ever might use RethrowFirst strategy
```

## License

- Author: Mark J. Olah
- Email: ([mjo@cs.unm.edu](mailto:mjo@cs.unm.edu) DOT edu)
- Copyright: 2019
- LICENSE: Apache 2.0. See [LICENSE](#) file.

## 4 Namespace Index

### 4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">mappel</a>	20
<a href="#">mappel::estimator</a>	47
<a href="#">mappel::estimator::subroutine</a>	51
<a href="#">mappel::mcmc</a>	54
<a href="#">mappel::methods</a>	
Templated functions for operating on a <a href="#">PointEmitterModel</a>	55
<a href="#">mappel::methods::debug</a>	67
<a href="#">mappel::methods::likelihood</a>	68

<a href="#">mappel::methods::likelihood::debug</a>	71
<a href="#">mappel::methods::objective</a>	72
<a href="#">mappel::methods::objective::debug</a>	78
<a href="#">mappel::methods::objective::openmp</a>	80
<a href="#">mappel::methods::openmp</a>	84
<a href="#">omp_exception_catcher</a>	90
<a href="#">omp_exception_catcher::impl_</a>	91

## 5 Hierarchical Index

### 5.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<a href="#">mappel::estimator::Estimator&lt; Model &gt;</a>	117
<a href="#">mappel::estimator::ThreadedEstimator&lt; Model &gt;</a>	840
<a href="#">mappel::estimator::CGaussHeuristicEstimator&lt; Model &gt;</a>	93
<a href="#">mappel::estimator::CGaussMLE&lt; Model &gt;</a>	105
<a href="#">mappel::estimator::HeuristicEstimator&lt; Model &gt;</a>	530
<a href="#">mappel::estimator::IterativeMaximizer&lt; Model &gt;</a>	556
<a href="#">mappel::estimator::NewtonDiagonalMaximizer&lt; Model &gt;</a>	699
<a href="#">mappel::estimator::NewtonMaximizer&lt; Model &gt;</a>	720
<a href="#">mappel::estimator::QuasiNewtonMaximizer&lt; Model &gt;</a>	784
<a href="#">mappel::estimator::TrustRegionMaximizer&lt; Model &gt;</a>	853
<a href="#">mappel::estimator::SimulatedAnnealingMaximizer&lt; Model &gt;</a>	805
<a href="#">mappel::ImageFormat1DBase</a>	543
<a href="#">mappel::Gauss1DModel</a>	184
<a href="#">mappel::Gauss1DMAP</a>	128
<a href="#">mappel::Gauss1DMLE</a>	156
<a href="#">mappel::Gauss2DsxyMAP</a>	480
<a href="#">mappel::Gauss1DsModel</a>	268
<a href="#">mappel::Gauss1DsMAP</a>	211

mappel::Gauss1DsMLE	239
mappel::PoissonNoise1DObjective	763
mappel::Gauss1DMAP	128
mappel::Gauss1DMLE	156
mappel::Gauss1DsMAP	211
mappel::Gauss1DsMLE	239
mappel::Gauss2DsxyMAP	480
ImageFormat1DBase	
PoissonGaussianNoise2DObjective< ModelBase >	760
mappel::ImageFormat2DBase	549
mappel::Gauss2DModel	355
mappel::Gauss2DMAP	295
mappel::Gauss2DMLE	325
mappel::Gauss2DsModel	449
mappel::Gauss2DsMAP	384
mappel::Gauss2DsMLE	417
mappel::Gauss2DsxyModel	507
mappel::PoissonNoise2DObjective	770
mappel::Gauss2DMAP	295
mappel::Gauss2DMLE	325
mappel::Gauss2DsMAP	384
mappel::Gauss2DsMLE	417
MappelError	
mappel::ArrayShapeError	91
mappel::ArraySizeError	92
mappel::LogicalError	577
mappel::ModelBoundsError	698
mappel::NotImplementedError	741
mappel::NumericalError	742
mappel::ParameterValueError	745

mappel::estimator::IterativeMaximizer< Model >::MaximizerData	592
mappel::MCMCAdaptorBase	674
mappel::MCMCAdaptor1D	601
mappel::Gauss1DModel	184
mappel::MCMCAdaptor1Ds	619
mappel::Gauss1DsModel	268
mappel::MCMCAdaptor2D	637
mappel::Gauss2DModel	355
mappel::MCMCAdaptor2Ds	655
mappel::Gauss2DsModel	449
mappel::mcmc::MCMCData	677
mappel::mcmc::MCMCDataStack	679
mappel::mcmc::MCMCDebugData	682
mappel::estimator::MLEData	47
mappel::estimator::MLEDataStack	47
mappel::estimator::MLEDebugData	47
omp_exception_catcher::impl_::OMPExcceptionCatcher< _dummy >	743
mappel::PointEmitterModel	746
mappel::Gauss1DModel	184
mappel::Gauss1DsModel	268
mappel::Gauss2DModel	355
mappel::Gauss2DsModel	449
mappel::Gauss2DsxyModel	507
mappel::MAPEstimator	577
mappel::Gauss1DMAP	128
mappel::Gauss1DsMAP	211
mappel::Gauss2DMAP	295
mappel::Gauss2DsMAP	384
mappel::Gauss2DsxyMAP	480
mappel::MCMCAdaptor1D	601

mappel::MLEstimator	683
mappel::Gauss1DMLE	156
mappel::Gauss1DsMLE	239
mappel::Gauss2DMLE	325
mappel::Gauss2DsMLE	417
mappel::estimator::ProfileBoundsData	777
mappel::estimator::ProfileBoundsDataStack	780
mappel::estimator::ProfileBoundsDebugData	47
mappel::estimator::ProfileLikelihoodData	47
mappel::Gauss2DsxyModel::Stencil	817
mappel::Gauss1DsModel::Stencil	822
mappel::Gauss2DModel::Stencil	826
mappel::Gauss2DsModel::Stencil	831
mappel::Gauss1DModel::Stencil	836

## 6 Class Index

### 6.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<b>mappel::ArrayShapeError</b> Array is not of the right dimensionality	91
<b>mappel::ArraySizeError</b> Array is not of the right size	92
<b>mappel::estimator::CGaussHeuristicEstimator&lt; Model &gt;</b>	93
<b>mappel::estimator::CGaussMLE&lt; Model &gt;</b>	105
<b>mappel::estimator::Estimator&lt; Model &gt;</b>	117
<b>mappel::Gauss1DMAP</b> A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	128
<b>mappel::Gauss1DMLE</b> A 1D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective	156
<b>mappel::Gauss1DModel</b> A base class for 1D Gaussian PSF with a fixed sigma (standard dev.)	184

<b>mappel::Gauss1DsMAP</b>	
A 1D Gaussian with variable PSF sigma under an Poisson read noise assumption and MAP Objective	211
<b>mappel::Gauss1DsMLE</b>	
A 1D Gaussian with variable PSF under an Poisson noise assumption and maximum-likelihood estimator	239
<b>mappel::Gauss1DsModel</b>	
Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels	268
<b>mappel::Gauss2DMAP</b>	
A 2D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	295
<b>mappel::Gauss2DMLE</b>	
A 2D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective	325
<b>mappel::Gauss2DModel</b>	
A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma	355
<b>mappel::Gauss2DsMAP</b>	
A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum a-posteriori objective	384
<b>mappel::Gauss2DsMLE</b>	
A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum-likelihood objective	417
<b>mappel::Gauss2DsModel</b>	
A base class for 2D Gaussian PSF where the gaussian sigma is controlled by a single scalar parameter which is called sigma_ratio. The size of the gaussian psf is sigma_ratio*psf_sigma, where psf_sigma is considered as a vector [psf_sigmaX, psf_sigmaY]	449
<b>mappel::Gauss2DsxyMAP</b>	
A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	480
<b>mappel::Gauss2DsxyModel</b>	
A base class for 2D Gaussian PSF with axis-aligned gaussian with free parameters for both sigma_x and sigma_y. Gaussian sigma parameters sigma_x and sigma_y are measured in units of pixels. The model has 6 parameters, [x,y,l,bg,sigma_x,sigma_y]	507
<b>mappel::estimator::HeuristicEstimator&lt; Model &gt;</b>	530
<b>mappel::ImageFormat1DBase</b>	
A virtual base class for 2D image localization objectives	543
<b>mappel::ImageFormat2DBase</b>	
A virtual base class for 2D image localization objectives	549
<b>mappel::estimator::IterativeMaximizer&lt; Model &gt;</b>	556
<b>mappel::LogicalError</b>	
Failure of code or algorithm logic	577



<a href="#">mappel::MAPEstimator</a>	
A Mixin class to configure a for MLE estimation (null prior)	577
<a href="#">mappel::estimator::IterativeMaximizer&lt; Model &gt;::MaximizerData</a>	592
<a href="#">mappel::MCMCAdaptor1D</a>	601
<a href="#">mappel::MCMCAdaptor1Ds</a>	619
<a href="#">mappel::MCMCAdaptor2D</a>	637
<a href="#">mappel::MCMCAdaptor2Ds</a>	655
<a href="#">mappel::MCMCAdaptorBase</a>	674
<a href="#">mappel::mcmc::MCMCData</a>	677
<a href="#">mappel::mcmc::MCMCDataStack</a>	679
<a href="#">mappel::mcmc::MCMCDebugData</a>	682
<a href="#">mappel::MLEstimator</a>	
A Mixin class to configure a for MLE estimation (null prior)	683
<a href="#">mappel::ModelBoundsError</a>	
Access outside the model bounds is attempted	698
<a href="#">mappel::estimator::NewtonDiagonalMaximizer&lt; Model &gt;</a>	699
<a href="#">mappel::estimator::NewtonMaximizer&lt; Model &gt;</a>	720
<a href="#">mappel::NotImplementedError</a>	
Feature not yet implemented	741
<a href="#">mappel::NumericalError</a>	
Expected numerical condition does not hold	742
<a href="#">omp_exception_catcher::impl_::OMPExcptionCatcher&lt; _dummy &gt;</a>	743
<a href="#">mappel::ParameterValueError</a>	
Parameter value is not valid	745
<a href="#">mappel::PointEmitterModel</a>	
A virtual Base type for point emitter localization models	746
<a href="#">PoissonGaussianNoise2DObjective&lt; ModelBase &gt;</a>	
A Base type for point emitter localization models that use 2d images	760
<a href="#">mappel::PoissonNoise1DObjective</a>	
A base class for 1D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel	763
<a href="#">mappel::PoissonNoise2DObjective</a>	
A base class for 2D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel	770

<a href="#">mappel::estimator::ProfileBoundsData</a>	777
<a href="#">mappel::estimator::ProfileBoundsDataStack</a>	780
<a href="#">mappel::estimator::QuasiNewtonMaximizer&lt; Model &gt;</a>	784
<a href="#">mappel::estimator::SimulatedAnnealingMaximizer&lt; Model &gt;</a>	805
<a href="#">mappel::Gauss2DsxyModel::Stencil</a> Stencil for 2D free-sigma (astigmatic) models	817
<a href="#">mappel::Gauss1DsModel::Stencil</a> Stencil for 1D variable-sigma models	822
<a href="#">mappel::Gauss2DModel::Stencil</a> Stencil for 2D fixed-sigma models	826
<a href="#">mappel::Gauss2DsModel::Stencil</a> Stencil for 2D scalar-sigma models	831
<a href="#">mappel::Gauss1DModel::Stencil</a> Stencil for 1D fixed-sigma models	836
<a href="#">mappel::estimator::ThreadedEstimator&lt; Model &gt;</a>	840
<a href="#">mappel::estimator::TrustRegionMaximizer&lt; Model &gt;</a>	853

## 7 File Index

### 7.1 File List

Here is a list of all files with brief descriptions:

<a href="#">display.cpp</a>	876
<a href="#">display.h</a> Textual image display with colors	877
<a href="#">estimator.cpp</a> Non-templated estimator helper routines and static constants	878
<a href="#">estimator.h</a> The class declaration and inline and templated functions for the Estimator class hierarchy	879
<a href="#">estimator_helpers.h</a> Estimator helper subroutines	882
<a href="#">estimator_impl.h</a>	883
<a href="#">estimator_statics.cpp</a>	884
<a href="#">Gauss1DMAP.cpp</a> The class definition and template Specializations for Gauss1DMAP	884

<a href="#">Gauss1DMAP.h</a>	885
The class declaration and inline and templated functions for Gauss1DMAP	
<a href="#">Gauss1DMLE.cpp</a>	886
The class definition and template Specializations for Gauss1DMLE	
<a href="#">Gauss1DMLE.h</a>	886
The class declaration and inline and templated functions for Gauss1DMLE	
<a href="#">Gauss1DModel.cpp</a>	887
The class definition and template Specializations for Gauss1DModel	
<a href="#">Gauss1DModel.h</a>	887
The class declaration and inline and templated functions for Gauss1DModel	
<a href="#">Gauss1DsMAP.cpp</a>	888
The class definition and template Specializations for Gauss1DsMAP	
<a href="#">Gauss1DsMAP.h</a>	889
The class declaration and inline and templated functions for Gauss1DsMAP	
<a href="#">Gauss1DsMLE.cpp</a>	889
The class definition and template Specializations for Gauss1DsMLE	
<a href="#">Gauss1DsMLE.h</a>	890
The class declaration and inline and templated functions for Gauss1DsMLE	
<a href="#">Gauss1DsModel.cpp</a>	891
The class definition and template Specializations for Gauss1DsModel	
<a href="#">Gauss1DsModel.h</a>	891
The class declaration and inline and templated functions for Gauss1DsModel	
<a href="#">Gauss2DMAP.cpp</a>	892
The class definition and template Specializations for Gauss2DMAP	
<a href="#">Gauss2DMAP.h</a>	892
The class declaration and inline and templated functions for Gauss2DMAP	
<a href="#">Gauss2DMLE.cpp</a>	893
The class definition and template Specializations for Gauss2DMLE	
<a href="#">Gauss2DMLE.h</a>	894
The class declaration and inline and templated functions for Gauss2DMLE	
<a href="#">Gauss2DModel.cpp</a>	894
The class definition and template Specializations for Gauss2DModel	
<a href="#">Gauss2DModel.h</a>	895
The class declaration and inline and templated functions for Gauss2DModel	
<a href="#">Gauss2DsMAP.cpp</a>	896
The class definition and template Specializations for Gauss2DsMAP	
<a href="#">Gauss2DsMAP.h</a>	896
The class declaration and inline and templated functions for Gauss2DsMAP	

<a href="#">Gauss2DsMLE.cpp</a>	
The class definition and template Specializations for Gauss2DsMLE	897
<a href="#">Gauss2DsMLE.h</a>	
The class declaration and inline and templated functions for Gauss2DsMLE	898
<a href="#">Gauss2DsModel.cpp</a>	
The class definition and template Specializations for Gauss2DsModel	898
<a href="#">Gauss2DsModel.h</a>	
The class declaration and inline and templated functions for Gauss2DsModel	899
<a href="#">Gauss2DsxyMAP.h</a>	
The class declaration and inline and templated functions for Gauss2DsxyMAP	900
<a href="#">Gauss2DsxyModel.h</a>	
The class declaration and inline and templated functions for Gauss2DsxyModel	901
<a href="#">ImageFormat1DBase.cpp</a>	
The class definition and template Specializations for ImageFormat1DBase	902
<a href="#">ImageFormat1DBase.h</a>	
The class declaration and inline and templated functions for ImageFormat1DBase	902
<a href="#">ImageFormat2DBase.cpp</a>	
The class definition and template Specializations for ImageFormat2DBase	903
<a href="#">ImageFormat2DBase.h</a>	
The class declaration and inline and templated functions for ImageFormat2DBase	904
<a href="#">MAPEstimator.h</a>	
Class declaration and inline and templated functions for MAPEstimator	905
<a href="#">mcmc.cpp</a>	
MCMC helper functions	906
<a href="#">mcmc.h</a>	
Templated MCMC methods for posterior estimation	907
<a href="#">mcmc_data.h</a>	
MCMC data storage types	908
<a href="#">MCMCAdaptor1D.cpp</a>	
The class definition and template Specializations for MCMCAdaptor1D	908
<a href="#">MCMCAdaptor1D.h</a>	
The class declaration and inline and templated functions for MCMCAdaptor1D	909
<a href="#">MCMCAdaptor1Ds.cpp</a>	
The class definition and template Specializations for MCMCAdaptor1Ds	909
<a href="#">MCMCAdaptor1Ds.h</a>	
The class declaration and inline and templated functions for MCMCAdaptor1Ds	910
<a href="#">MCMCAdaptor2D.cpp</a>	
The class definition and template Specializations for MCMCAdaptor2D	911

<a href="#">MCMCAdaptor2D.h</a>	
The class declaration and inline and templated functions for MCMCAdaptor2D	911
<a href="#">MCMCAdaptor2Ds.cpp</a>	
The class definition and template Specializations for MCMCAdaptor2Ds	912
<a href="#">MCMCAdaptor2Ds.h</a>	
The class declaration and inline and templated functions for MCMCAdaptor2Ds	912
<a href="#">MCMCAdaptorBase.cpp</a>	
The class definition and template Specializations for MCMCAdaptorBase	913
<a href="#">MCMCAdaptorBase.h</a>	
The class declaration and inline and templated functions for MCMCAdaptorBase	913
<a href="#">MLEstimator.h</a>	
Class declaration and inline and templated functions for MLEstimator	914
<a href="#">model_methods.h</a>	915
<a href="#">model_methods_impl.h</a>	918
<a href="#">numerical.cpp</a>	
Numerical matrix operations	921
<a href="#">numerical.h</a>	
Numerical matrix operations	922
<a href="#">OMPExceptionCatcher.h</a>	
A lightweight class for managing C++ exception handling strategies for OpenMP methods	923
<a href="#">openmp_methods.h</a>	
Namespaces for OpenMP parallelized versions of the mappel::model namespace functions (external methods)	925
<a href="#">PointEmitterModel.cpp</a>	
The class definition and template Specializations for PointEmitterModel	927
<a href="#">PointEmitterModel.h</a>	
The class declaration and inline and templated functions for PointEmitterModel	928
<a href="#">PoissonGaussianNoise2DObjective.cpp</a>	929
<a href="#">PoissonGaussianNoise2DObjective.h</a>	
The class declaration and inline and templated functions for <a href="#">PoissonGaussianNoise2DObjective</a>	929
<a href="#">PoissonNoise1DObjective.cpp</a>	
The class definition and template Specializations for PoissonNoise1DObjective	932
<a href="#">PoissonNoise1DObjective.h</a>	
The class declaration and inline and templated functions for PoissonNoise1DObjective	933
<a href="#">PoissonNoise2DObjective.cpp</a>	
The class definition and template Specializations for PoissonNoise2DObjective	935
<a href="#">PoissonNoise2DObjective.h</a>	
The class declaration and inline and templated functions for PoissonNoise2DObjective	935

<a href="#">rng.cpp</a>	
Global random number generator	937
<a href="#">rng.h</a>	
Random number generation using sfmt	938
<a href="#">stencil.cpp</a>	
The stencils for pixel based computations	939
<a href="#">stencil.h</a>	
The stencils for pixel based computations	940
<a href="#">util.cpp</a>	941
<a href="#">util.h</a>	
Common utilities and errors	942

## 8 Namespace Documentation

### 8.1 mappel Namespace Reference

#### Namespaces

- [estimator](#)
- [mcmc](#)
- [methods](#)

*Templated functions for operating on a [PointEmitterModel](#).*

#### Classes

- struct [ArrayShapeError](#)  
*Array is not of the right dimensionality.*
- struct [ArraySizeError](#)  
*Array is not of the right size.*
- class [Gauss1DMAP](#)  
*A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.*
- class [Gauss1DMLE](#)  
*A 1D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective.*
- class [Gauss1DModel](#)  
*A base class for 1D Gaussian PSF with a fixed sigma (standard dev.)*
- class [Gauss1DsMAP](#)  
*A 1D Gaussian with variable PSF sigma under an Poisson read noise assumption and MAP Objective.*
- class [Gauss1DsMLE](#)  
*A 1D Gaussian with variable PSF under an Poisson noise assumption and maximum-likelihood estimator.*
- class [Gauss1DsModel](#)  
*Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels.*
- class [Gauss2DMAP](#)

- A 2D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.*

  - class [Gauss2DMLE](#)
- A 2D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective.*

  - class [Gauss2DModel](#)
- A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma.*

  - class [Gauss2DsMAP](#)
- A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum a-posteriori objective.*

  - class [Gauss2DsMLE](#)
- A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum-likelihood objective.*

  - class [Gauss2DsModel](#)
- A base class for 2D Gaussian PSF where the gaussian sigma is controlled by a single scalar parameter which is called sigma\_ratio. The size of the gaussian psf is sigma\_ratio\*psf\_sigma, where psf\_sigma is considered as a vector [psf\_sigmaX, psf\_sigmaY].*

  - class [Gauss2DsxyMAP](#)
- A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.*

  - class [Gauss2DsxyModel](#)
- A base class for 2D Gaussian PSF with axis-aligned gaussian with free parameters for both sigma\_x and sigma\_y. Gaussian sigma parameters sigma\_x and sigma\_y are measured in units of pixels. The model has 6 parameters, [x,y,l,bg,sigma\_x,sigma\_y].*

  - class [ImageFormat1DBase](#)
- A virtual base class for 2D image localization objectives.*

  - class [ImageFormat2DBase](#)
- A virtual base class for 2D image localization objectives.*

  - struct [LogicalError](#)
- Failure of code or algorithm logic.*

  - class [MAPEstimator](#)
- A Mixin class to configure a for MLE estimation (null prior).*

  - class [MCMCAdaptor1D](#)
  - class [MCMCAdaptor1Ds](#)
  - class [MCMCAdaptor2D](#)
  - class [MCMCAdaptor2Ds](#)
  - class [MCMCAdaptorBase](#)
  - class [MLEstimator](#)
- A Mixin class to configure a for MLE estimation (null prior).*

  - struct [ModelBoundsError](#)
- Access outside the model bounds is attempted.*

  - struct [NotImplementedError](#)
- Feature not yet implemented.*

  - struct [NumericalError](#)
- Expected numerical condition does not hold.*

  - struct [ParameterValueError](#)
- Parameter value is not valid.*

  - class [PointEmitterModel](#)
- A virtual Base type for point emitter localization models.*

  - class [PoissonNoise1DObjective](#)
- A base class for 1D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel.*

- class [PoissonNoise2DObjective](#)

*A base class for 2D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel.*

## Typedefs

- using [ParallelRngGeneratorT](#) = `trng::lcg64_shift`
- using [ParallelRngManagerT](#) = `parallel_rng::ParallelRngManager< ParallelRngGeneratorT >`
- using [RngSeedT](#) = `parallel_rng::SeedT`
- using [UniformDistT](#) = `std::uniform_real_distribution< double >`
- using [BoolT](#) = `uint16_t`
- using [BoolVecT](#) = `arma::Col< uint16_t >`
- using [IdxT](#) = `arma::uword`
- using [IdxVecT](#) = `arma::Col< IdxT >`
- using [IdxMatT](#) = `arma::Mat< IdxT >`
- using [VecT](#) = `arma::vec`
- using [MatT](#) = `arma::mat`
- using [CubeT](#) = `arma::cube`
- using [VecFieldT](#) = `arma::field< VecT >`
- using [StatsT](#) = `std::map< std::string, double >`
- using [StringVecT](#) = `std::vector< std::string >`
- template<class ModelT, class ModelBaseT >  
using [EnableIfSubclassT](#) = `typename std::enable_if< std::is_base_of< ModelBaseT, ModelT >::value, void >::type`
- template<class ReturnT, class ModelT, class ModelBaseT >  
using [ReturnIfSubclassT](#) = `typename std::enable_if< std::is_base_of< ModelBaseT, ModelT >::value, ReturnT >::type`
- template<class Model >  
using [ImageCoordT](#) = `typename Model::ImageCoordT`
- template<class Model >  
using [ImagePixelT](#) = `typename Model::ImagePixelT`
- template<class Model >  
using [ParamT](#) = `typename Model::ParamT`
- template<class Model >  
using [ParamVecT](#) = `typename Model::ParamVecT`
- template<class Model >  
using [ImageT](#) = `typename Model::ImageT`
- template<class Model >  
using [ModelDataT](#) = `typename Model::ModelDataT`
- template<class Model >  
using [StencilT](#) = `typename Model::Stencil`
- template<class Model >  
using [ImageStackT](#) = `typename Model::ImageStackT`
- template<class Model >  
using [ModelDataStackT](#) = `typename Model::ModelDataStackT`
- template<class Model >  
using [StencilVecT](#) = `typename Model::StencilVecT`
- using [MappelError](#) = `backtrace_exception::BacktraceException`



## Functions

- `const char * lambda\_term\_color (int size, int Lidx)`
- `ostream & print\_centered\_title (ostream &out, char fill, int width, const char *title=nullptr)`
- `ostream & print\_labeled\_image (ostream &out, const arma::mat &im, const char *title, const char *color)`
- `template<>  
std::ostream & print\_image (std::ostream &out, const arma::vec &im)`
- `template<>  
std::ostream & print\_image (std::ostream &out, const arma::mat &im)`
- `template<>  
std::ostream & print\_text\_image (std::ostream &out, const arma::vec &im)`
- `template<>  
std::ostream & print\_text\_image (std::ostream &out, const arma::mat &im)`
- `template<>  
std::ostream & print\_image (std::ostream &out, const arma::cube &im)`
- `std::ostream & operator<< (std::ostream &out, const Gauss1DModel::Stencil &s)`
- `std::ostream & operator<< (std::ostream &out, const Gauss1DsModel::Stencil &s)`
- `std::ostream & operator<< (std::ostream &out, const Gauss2DModel::Stencil &s)`
- `std::ostream & operator<< (std::ostream &out, const Gauss2DsModel::Stencil &s)`
- `void copy\_Usym\_mat (arma::mat &usym)`
- `void copy\_Usym\_mat\_stack (arma::cube &usym_stack)`
- `void copy\_Lsym\_mat (arma::mat &lsym)`
- `void cholesky\_make\_negative\_definite (arma::mat &m)`
- `void cholesky\_make\_positive\_definite (arma::mat &m)`
- `bool is\_negative\_definite (const arma::mat &usym)`
- `bool is\_positive\_definite (const arma::mat &usym)`
- `bool is\_symmetric (const arma::mat &A)`
- `void cholesky\_convert\_lower\_triangular (arma::mat &chol)`
- `void cholesky\_convert\_full\_matrix (arma::mat &chol)`
- `bool cholesky (arma::mat &A)`
- `bool modified\_cholesky (arma::mat &A)`
- `arma::vec cholesky\_solve (const arma::mat &C, const arma::vec &b)`
- `double norm\_sq (const VecT &v)`
- `double normal\_quantile\_twosided (double confidence)`
- `double normal\_quantile\_onesided (double confidence)`
- `double chisq\_quantile (double confidence, int dof)`
- `double chisq\_quantile (double confidence)`
- `void fill\_gaussian\_stencil (int size, double stencil[], double sigma)`
- `double gaussian\_convolution (int x, int y, const MatT &data, const VecT &Xstencil, const VecT &Ystencil)`
- `void estimate\_gaussian\_2Dmax (const MatT &data, const VecT &Xstencil, const VecT &Ystencil, int max_pos[], double &min_val)`
- `void refine\_gaussian\_2Dmax (const MatT &data, const VecT &Xstencil, const VecT &Ystencil, int max_pos[])`
- `double gaussian\_3D\_convolution (int x, int y, int z, const CubeT &data, const VecFieldT &stencils)`
- `void estimate\_gaussian\_3Dmax (const CubeT &data, const VecFieldT &stencils, int max_pos[], double &min_val)`
- `void refine\_gaussian\_3Dmax (const CubeT &data, const VecFieldT &stencils, int max_pos[])`
- `double estimate\_background (const MatT &im, const MatT &unit_model_im, double min_bg)`
- `double estimate\_intensity (const MatT &im, const MatT &unit_model_im, double bg)`
- `double estimate\_background (const CubeT &im, const CubeT &unit_model_im)`
- `double estimate\_intensity (const CubeT &im, const CubeT &unit_model_im, double bg)`
- `void enable\_all\_cpus ()`
- `bool istarts\_with (const char *s, const char *pattern)`

- bool [istarts\\_with](#) (const std::string &str, const char \*pattern)
- const char \* [icontains](#) (const char \*s, const char \*pattern)
- int [maxidx](#) (const [VecT](#) &v)
- std::ostream & [operator<<](#) (std::ostream &out, const [StatsT](#) &stats)
- template<class ImageT >  
std::ostream & [print\\_image](#) (std::ostream &out, const [ImageT](#) &im)
- template<class ImageT >  
std::ostream & [print\\_text\\_image](#) (std::ostream &out, const [ImageT](#) &im)
- template<class Vec >  
std::ostream & [print\\_vec\\_row](#) (std::ostream &out, const Vec &vec, const char \*header, int header\_width, const char \*color=nullptr)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[heuristic\\_compute\\_estimate](#) (const Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model >  
> &theta\_init)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[compute\\_estimate](#) (Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta\_init,  
int max\_iterations)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[compute\\_estimate\\_debug](#) (const Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model >  
&theta\_init, int max\_iterations, [ParamVecT](#)< Model > &sequence)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DsModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[heuristic\\_compute\\_estimate](#) (const Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model >  
> &theta\_init)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DsModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[compute\\_estimate](#) (Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta\_init,  
int max\_iterations)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DsModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[compute\\_estimate\\_debug](#) (const Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model >  
&theta\_init, int max\_iterations, [ParamVecT](#)< Model > &sequence)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DsxyModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[heuristic\\_compute\\_estimate](#) (const Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model >  
&theta\_init)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DsxyModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[compute\\_estimate](#) (Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta\_init, int  
max\_iterations)
- template<class Model >  
std::enable\_if< std::is\_base\_of< [Gauss2DsxyModel](#), Model >::value, [ParamT](#)< Model > >::type [cgauss\\_↵](#)  
[compute\\_estimate\\_debug](#) (const Model &model, const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model >  
&theta\_init, int max\_iterations, [ParamVecT](#)< Model > &sequence)
- template<class FloatT >  
FloatT [clamp](#) (FloatT val, FloatT min\_val, FloatT max\_val)
- template<class Model, typename = EnableIfSubclassT<Model,PointEmitterModel>>  
std::ostream & [operator<<](#) (std::ostream &out, const Model &model)
- template<class RngT >  
[IdxT generate\\_poisson\\_small](#) (RngT &rng, double mu)

*Generates a single Poisson distributed int from distribution with mean mu.*

- `template<class RngT >`  
`IdxT generate_poisson_large` (RngT &rng, double mu)
- `template<class RngT >`  
`double generate_poisson` (RngT &rng, double mu)
- `double gauss_norm` (double sigma)
- `void fill_d_stencil` (int size, double stencil[], double theta\_x)
- `void fill_G_stencil` (int size, double stencil[], const double dx[], double theta\_sigma)
- `void fill_X_stencil` (int size, double stencil[], const double dx[], double theta\_sigma)
- `void fill_DX_stencil` (int size, double stencil[], const double Gx[], double theta\_sigma)
- `void fill_DXS_stencil` (int size, double stencil[], const double dx[], const double Gx[], double theta\_sigma)
- `void fill_DXS2_stencil` (int size, double stencil[], const double dx[], const double Gx[], const double DXS[], double theta\_sigma)
- `void fill_DXSX_stencil` (int size, double stencil[], const double dx[], const double Gx[], const double DX[], double theta\_sigma)
- `VecT make_d_stencil` (int size, double theta\_x)
- `VecT make_G_stencil` (int size, const VecT &dx, double theta\_sigma)
- `VecT make_X_stencil` (int size, const VecT &dx, double theta\_sigma)
- `VecT make_DX_stencil` (int size, const VecT &Gx, double theta\_sigma)
- `VecT make_DXS_stencil` (int size, const VecT &dx, const VecT &Gx, double theta\_sigma)
- `VecT make_DXS2_stencil` (int size, const VecT &dx, const VecT &Gx, const VecT &DXS, double theta\_sigma)
- `VecT make_DXSX_stencil` (int size, const VecT &dx, const VecT &Gx, const VecT &DX, double theta\_sigma)
- `VecT make_gaussian_stencil` (int size, double sigma)
- `double poisson_log_likelihood` (double model\_val, double data\_val)
- `double relative_poisson_log_likelihood` (double model\_val, double data\_val)
- `double check_lower_bound_hyperparameter` (const char \*name, double value, double lower\_bound)
- `double check_positive_hyperparameter` (const char \*name, double value, double hyperprior\_epsilon=1E-6)
- `double check_unit_hyperparameter` (const char \*name, double value, double hyperprior\_epsilon=1E-6)
- `double log_prior_beta_const` (double beta)
- `double log_prior_beta2_const` (double beta0, double beta1)
- `double log_prior_gamma_const` (double kappa, double mean)
- `double log_prior_pareto_const` (double alpha, double min)
- `double log_prior_normal_const` (double sigma)
- `double rllh_beta_prior` (double beta, double v, double max=1., double min=0.)
- `double rllh_beta2_prior` (double beta0, double beta1, double v, double max=1., double min=0.)
- `double rllh_gamma_prior` (double kappa, double mean, double v)
- `double rllh_pareto_prior` (double alpha, double v)
- `double rllh_normal_prior` (double mu, double sigma)
- `double beta_prior_grad` (double beta, double v, double max=1., double min=0.)
- `double beta2_prior_grad` (double beta0, double beta1, double v, double max=1., double min=0.)
- `double gamma_prior_grad` (double kappa, double mean, double v)
- `double pareto_prior_grad` (double alpha, double v)
- `double normal_prior_grad` (double mu, double sigma)
- `double beta_prior_grad2` (double beta, double v, double max=1., double min=0.)
- `double beta2_prior_grad2` (double beta0, double beta1, double v, double max=1., double min=0.)
- `double gamma_prior_grad2` (double kappa, double v)
- `double pareto_prior_grad2` (double alpha, double v)
- `double normal_prior_grad` (double sigma)
- `double rllh_normal_prior` (double mu, double sigma, double v)
- `double normal_prior_grad` (double mu, double sigma, double v)
- `double normal_prior_grad2` (double sigma)

- `template<typename T >`  
`int sgn (T val)`  
*sign (signum) function: -1/0/1*
- `template<typename T >`  
`T square (T x)`
- `double restrict\_value\_range (double val, double minval, double maxval)`
- `template<typename T , typename... Args>`  
`std::unique_ptr< T > make\_unique (Args &&...args)`

## Variables

- `const char * TERM\_BLACK ="1;30"`
- `const char * TERM\_RED ="1;31"`
- `const char * TERM\_GREEN ="1;32"`
- `const char * TERM\_YELLOW ="1;33"`
- `const char * TERM\_BLUE ="1;34"`
- `const char * TERM\_MAGENTA ="1;35"`
- `const char * TERM\_CYAN ="1;36"`
- `const char * TERM\_WHITE ="1;37"`
- `const char * TERM\_DIM\_BLACK ="0;30"`
- `const char * TERM\_DIM\_RED ="0;31"`
- `const char * TERM\_DIM\_GREEN ="0;32"`
- `const char * TERM\_DIM\_YELLOW ="0;33"`
- `const char * TERM\_DIM\_BLUE ="0;34"`
- `const char * TERM\_DIM\_MAGENTA ="0;35"`
- `const char * TERM\_DIM\_CYAN ="0;36"`
- `const char * TERM\_DIM\_WHITE ="0;37"`
- `ParallelRngManagerT rng_manager`

### 8.1.1 Detailed Description

All models will call for maximization through this virtual function. All non-GPU based maximizers will use this version which spawns threads using a non-virtual entry point member function `Maximizer::thread_entry`. GPU-based maximizers will want to do something custom, so they will declare their own virtual `maximize_stack`.

It is also because of the GPU-based maximizers that we are putting initialization, and CRLB/LLH calculations in here even though the Model knows how to do them.

We expect that those methods will need to also be parallelized and the GPU will need custom code, and the threaded CPU versions will want to also compute those in parallel, so in order to have a consistent call interface to the `Maximizer` classes, we put the CRLB/LLH and initialization work within the `maximize_stack` method.

### 8.1.2 Typedef Documentation

#### 8.1.2.1 using `mappel::BoolT = typedef uint16_t`

Definition at line 23 of file `util.h`.

**8.1.2.2** `using mappel::BoolVecT = typedef arma::Col<uint16_t>`

Definition at line 24 of file util.h.

**8.1.2.3** `using mappel::CubeT = typedef arma::cube`

A type to represent floating-point data cubes

Definition at line 30 of file util.h.

**8.1.2.4** `template<class ModelT , class ModelBaseT > using mappel::EnableIfSubclassT = typedef typename std::enable_if<std::is_base_of<ModelBaseT,ModelT>::value,void>::type`

Definition at line 37 of file util.h.

**8.1.2.5** `using mappel::IdxMatT = typedef arma::Mat<IdxT>`

A type to represent integer data arrays

Definition at line 27 of file util.h.

**8.1.2.6** `using mappel::IdxT = typedef arma::uword`

Definition at line 25 of file util.h.

**8.1.2.7** `using mappel::IdxVecT = typedef arma::Col<IdxT>`

A type to represent integer data arrays

Definition at line 26 of file util.h.

**8.1.2.8** `template<class Model > using mappel::ImageCoordT = typedef typename Model::ImageCoordT`

Definition at line 42 of file util.h.

**8.1.2.9** `template<class Model > using mappel::ImagePixelT = typedef typename Model::ImagePixelT`

Definition at line 43 of file util.h.

**8.1.2.10** `template<class Model > using mappel::ImageStackT = typedef typename Model::ImageStackT`

Definition at line 51 of file util.h.

**8.1.2.11** `template<class Model > using mappel::ImageT = typedef typename Model::ImageT`

Definition at line 47 of file util.h.

**8.1.2.12** `using mappel::MappelError = typedef backtrace_exception::BacktraceException`

Definition at line 64 of file util.h.

**8.1.2.13** `using mappel::MatT = typedef arma::mat`

A type to represent floating-point data matrices

Definition at line 29 of file util.h.

**8.1.2.14** `template<class Model> using mappel::ModelDataStackT = typedef typename Model::ModelDataStackT`

Definition at line 52 of file util.h.

**8.1.2.15** `template<class Model> using mappel::ModelDataT = typedef typename Model::ModelDataT`

Definition at line 48 of file util.h.

**8.1.2.16** `using mappel::ParallelRngGeneratorT = typedef trng::lcg64_shift`

Definition at line 21 of file rng.h.

**8.1.2.17** `using mappel::ParallelRngManagerT = typedef parallel_rng::ParallelRngManager<ParallelRngGeneratorT>`

Definition at line 22 of file rng.h.

**8.1.2.18** `template<class Model> using mappel::ParamT = typedef typename Model::ParamT`

Definition at line 45 of file util.h.

**8.1.2.19** `template<class Model> using mappel::ParamVecT = typedef typename Model::ParamVecT`

Definition at line 46 of file util.h.

**8.1.2.20** `template<class ReturnT, class ModelT, class ModelBaseT> using mappel::ReturnIfSubclassT = typedef typename std::enable_if<std::is_base_of<ModelBaseT, ModelT>::value, ReturnT>::type`

Definition at line 40 of file util.h.

**8.1.2.21** `using mappel::RngSeedT = typedef parallel_rng::SeedT`

Definition at line 23 of file rng.h.

**8.1.2.22** `using mappel::StatsT = typedef std::map<std::string, double>`

A convenient form for reporting dictionaries of named FP data to Matlab

Definition at line 32 of file util.h.

8.1.2.23 `template<class Model > using mappel::StencilT = typedef typename Model::Stencil`

Definition at line 49 of file util.h.

8.1.2.24 `template<class Model > using mappel::StencilVecT = typedef typename Model::StencilVecT`

Definition at line 53 of file util.h.

8.1.2.25 `using mappel::StringVecT = typedef std::vector<std::string>`

Definition at line 33 of file util.h.

8.1.2.26 `using mappel::UniformDistT = typedef std::uniform_real_distribution<double>`

Definition at line 24 of file rng.h.

8.1.2.27 `using mappel::VecFieldT = typedef arma::field<VecT>`

Definition at line 31 of file util.h.

8.1.2.28 `using mappel::VecT = typedef arma::vec`

A type to represent floating-point data arrays

Definition at line 28 of file util.h.

### 8.1.3 Function Documentation

8.1.3.1 `double mappel::beta2_prior_grad ( double beta0, double beta1, double v, double max = 1 . , double min = 0 . )`  
[inline]

Definition at line 316 of file stencil.h.

8.1.3.2 `double mappel::beta2_prior_grad2 ( double beta0, double beta1, double v, double max = 1 . , double min = 0 . )`  
[inline]

Definition at line 349 of file stencil.h.

8.1.3.3 `double mappel::beta_prior_grad ( double beta, double v, double max = 1 . , double min = 0 . )` [inline]

Definition at line 309 of file stencil.h.

8.1.3.4 `double mappel::beta_prior_grad2 ( double beta, double v, double max = 1 . , double min = 0 . )` [inline]

Definition at line 341 of file stencil.h.

8.1.3.5 `template<class Model > std::enable_if<std::is_base_of<Gauss2DModel,Model>::value, ParamT<Model> >::type mappel::cgauss_compute_estimate ( Model & model, const ModelDataT< Model > & im, const ParamT< Model > & theta_init, int max_iterations )`

Definition at line 223 of file Gauss2DModel.h.

References mappel::Gauss2DModel::psf\_sigma, and mappel::ImageFormat2DBase::size.

Referenced by mappel::estimator::CGaussMLE< Model >::get\_debug\_stats().

8.1.3.6 `template<class Model > std::enable_if<std::is_base_of<Gauss2DsxyModel,Model>::value, ParamT<Model> >::type mappel::cgauss_compute_estimate ( Model & model, const ModelDataT< Model > & im, const ParamT< Model > & theta_init, int max_iterations )`

Definition at line 251 of file Gauss2DsxyModel.h.

References mappel::ImageFormat2DBase::size.

8.1.3.7 `template<class Model > std::enable_if<std::is_base_of<Gauss2DsModel,Model>::value, ParamT<Model> >::type mappel::cgauss_compute_estimate ( Model & model, const ModelDataT< Model > & im, const ParamT< Model > & theta_init, int max_iterations )`

Definition at line 253 of file Gauss2DsModel.h.

References mappel::ImageFormat2DBase::size.

8.1.3.8 `template<class Model > std::enable_if<std::is_base_of<Gauss2DModel,Model>::value, ParamT<Model> >::type mappel::cgauss_compute_estimate_debug ( const Model & model, const ModelDataT< Model > & im, const ParamT< Model > & theta_init, int max_iterations, ParamVecT< Model > & sequence )`

Definition at line 238 of file Gauss2DModel.h.

References mappel::Gauss2DModel::psf\_sigma, and mappel::ImageFormat2DBase::size.

Referenced by mappel::estimator::CGaussMLE< Model >::get\_debug\_stats().

8.1.3.9 `template<class Model > std::enable_if<std::is_base_of<Gauss2DsxyModel,Model>::value, ParamT<Model> >::type mappel::cgauss_compute_estimate_debug ( const Model & model, const ModelDataT< Model > & im, const ParamT< Model > & theta_init, int max_iterations, ParamVecT< Model > & sequence )`

Definition at line 266 of file Gauss2DsxyModel.h.

References mappel::ImageFormat2DBase::size.

8.1.3.10 `template<class Model > std::enable_if<std::is_base_of<Gauss2DsModel,Model>::value, ParamT<Model> >::type mappel::cgauss_compute_estimate_debug ( const Model & model, const ModelDataT< Model > & im, const ParamT< Model > & theta_init, int max_iterations, ParamVecT< Model > & sequence )`

Definition at line 268 of file Gauss2DsModel.h.

References mappel::ImageFormat2DBase::size.



8.1.3.11 `template<class Model> std::enable_if<std::is_base_of<Gauss2DModel,Model>::value,ParamT<Model>>::type  
mappel::cgauss_heuristic_compute_estimate ( const Model & model, const ModelDataT<Model> & im, const  
ParamT<Model> & theta_init )`

Definition at line 209 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::psf_sigma`, and `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::estimator::HeuristicEstimator<Model>::get_debug_stats()`, and `mappel::estimator::CGaussMLE<Model>::get_debug_stats()`.

8.1.3.12 `template<class Model> std::enable_if<std::is_base_of<Gauss2DsxyModel,Model>::value,ParamT<Model>>::type  
mappel::cgauss_heuristic_compute_estimate ( const Model & model, const ModelDataT<Model> & im, const  
ParamT<Model> & theta_init )`

Definition at line 237 of file Gauss2DsxyModel.h.

References `mappel::ImageFormat2DBase::size`.

8.1.3.13 `template<class Model> std::enable_if<std::is_base_of<Gauss2DsModel,Model>::value,ParamT<Model>>::type  
mappel::cgauss_heuristic_compute_estimate ( const Model & model, const ModelDataT<Model> & im, const  
ParamT<Model> & theta_init )`

Definition at line 239 of file Gauss2DsModel.h.

References `mappel::ImageFormat2DBase::size`.

8.1.3.14 `double mappel::check_lower_bound_hyperparameter ( const char * name, double value, double lower_bound )`

8.1.3.15 `double mappel::check_positive_hyperparameter ( const char * name, double value, double hyperprior_epsilon = 1E-6 )`

8.1.3.16 `double mappel::check_unit_hyperparameter ( const char * name, double value, double hyperprior_epsilon = 1E-6 )`

8.1.3.17 `double mappel::chisq_quantile ( double confidence, int dof )`

Definition at line 43 of file stencil.cpp.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`, `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`, and `mappel::methods::openmp::error_bounds_profile_likelihood_stack()`.

8.1.3.18 `double mappel::chisq_quantile ( double confidence )`

Definition at line 50 of file stencil.cpp.

8.1.3.19 `bool mappel::cholesky ( arma::mat & A )`

Convert full or upper-triangular symmetric matrix to lower-triangular Cholesky decomposition in-place

No error checking is performed

## Parameters

in, out		
---------	--	--

Definition at line 102 of file numerical.cpp.

Referenced by `is_positive_definite()`, `mappel::estimator::subroutine::solve_restricted_step_length_newton()`, and `mappel::estimator::subroutine::solve_TR_subproblem()`.

#### 8.1.3.20 `void mappel::cholesky_convert_full_matrix ( arma::mat & chol )`

Convert matrix in internal Cholesky format into a full matrix  $M = L * L'$

Definition at line 82 of file numerical.cpp.

References `copy_Usym_mat()`.

Referenced by `cholesky_make_negative_definite()`, and `cholesky_make_positive_definite()`.

#### 8.1.3.21 `void mappel::cholesky_convert_lower_triangular ( arma::mat & chol )`

Convert matrix in internal Cholesky format into a lower triangular matrix  $L$  where  $M = L * L'$

Definition at line 71 of file numerical.cpp.

Referenced by `mappel::estimator::subroutine::solve_restricted_step_length_newton()`.

#### 8.1.3.22 `void mappel::cholesky_make_negative_definite ( arma::mat & m )`

Modify  $m$  in-place using modified Cholesky decomposition to ensure  $m$  is negative definite

Definition at line 38 of file numerical.cpp.

References `cholesky_convert_full_matrix()`, and `modified_cholesky()`.

Referenced by `mappel::methods::objective::negative_definite_hessian()`.

#### 8.1.3.23 `void mappel::cholesky_make_positive_definite ( arma::mat & m )`

Modify  $m$  in-place using modified Cholesky decomposition to ensure  $m$  is positive definite

Definition at line 46 of file numerical.cpp.

References `cholesky_convert_full_matrix()`, and `modified_cholesky()`.

#### 8.1.3.24 `arma::vec mappel::cholesky_solve ( const arma::mat & C, const arma::vec & b )`

Given a matrix in modified Cholesky format and a vector solve the linear system  $C x = b$ .

## Parameters

<i>C</i>	A matrix in lower modified Cholesky format
<i>b</i>	A vector representing the right hand side of the linear system.

## Returns

*x* - the solution to the linear system

Definition at line 186 of file numerical.cpp.

Referenced by `mappel::estimator::subroutine::solve_restricted_step_length_newton()`, and `mappel::estimator::subroutine::solve_TR_subproblem()`.

**8.1.3.25** `template<class FloatT > FloatT mappel::clamp ( FloatT val, FloatT min_val, FloatT max_val )`

Definition at line 103 of file numerical.h.

References `norm_sq()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::subroutine::compute_D_scale()`, `mappel::estimator::subroutine::compute_initial_trust_radius()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::subroutine::solve_restricted_step_length_newton()`.

**8.1.3.26** `void mappel::copy_Lsym_mat ( arma::mat & lsym )`

Convert symmetric matrix stored as lower triangular to full Matrix Assuming *lsym* is the main diagonal and lower triangle of a symmetric matrix, fill in the upper triangle by copying the lower triangle. This operation modifies the matrix.

Definition at line 30 of file numerical.cpp.

**8.1.3.27** `void mappel::copy_USym_mat ( arma::mat & usym )`

Convert symmetric matrix stored as upper triangular to full Matrix Assuming *usym* is the main diagonal and upper triangle of a symmetric matrix, fill in the lower triangle by copying the upper triangle. This operation modifies the matrix.

Definition at line 13 of file numerical.cpp.

Referenced by `cholesky_convert_full_matrix()`.

**8.1.3.28** `void mappel::copy_USym_mat_stack ( arma::cube & usym_stack )`

Definition at line 20 of file numerical.cpp.

**8.1.3.29** `void mappel::enable_all_cpus ( )`

Definition at line 16 of file util.cpp.

8.1.3.30 `double mappel::estimate_background ( const MatT & im, const MatT & unit_model_im, double min_bg )`

Definition at line 280 of file stencil.cpp.

8.1.3.31 `double mappel::estimate_background ( const CubeT & im, const CubeT & unit_model_im )`

Definition at line 299 of file stencil.cpp.

8.1.3.32 `void mappel::estimate_gaussian_2Dmax ( const MatT & data, const VecT & Xstencil, const VecT & Ystencil, int max_pos[], double & min_val )`

Definition at line 158 of file stencil.cpp.

References `gaussian_convolution()`.

8.1.3.33 `void mappel::estimate_gaussian_3Dmax ( const CubeT & data, const VecFieldT & stencils, int max_pos[], double & min_val )`

Definition at line 222 of file stencil.cpp.

References `gaussian_3D_convolution()`.

8.1.3.34 `double mappel::estimate_intensity ( const MatT & im, const MatT & unit_model_im, double bg )`

Definition at line 294 of file stencil.cpp.

8.1.3.35 `double mappel::estimate_intensity ( const CubeT & im, const CubeT & unit_model_im, double bg )`

Definition at line 309 of file stencil.cpp.

8.1.3.36 `void mappel::fill_d_stencil ( int size, double stencil[], double theta_x ) [inline]`

Definition at line 153 of file stencil.h.

Referenced by `make_d_stencil()`.

8.1.3.37 `void mappel::fill_DX_stencil ( int size, double stencil[], const double Gx[], double theta_sigma ) [inline]`

Definition at line 178 of file stencil.h.

Referenced by `make_DX_stencil()`.

8.1.3.38 `void mappel::fill_DXS2_stencil ( int size, double stencil[], const double dx[], const double Gx[], const double DXS[], double theta_sigma ) [inline]`

Definition at line 192 of file stencil.h.

Referenced by `make_DXS2_stencil()`.

8.1.3.39 `void mappel::fill_DXS_stencil ( int size, double stencil[], const double dx[], const double Gx[], double theta_sigma ) [inline]`

Definition at line 185 of file stencil.h.

References `square()`.

Referenced by `make_DXS_stencil()`.

8.1.3.40 `void mappel::fill_DXSX_stencil ( int size, double stencil[], const double dx[], const double Gx[], const double DX[], double theta_sigma ) [inline]`

Definition at line 205 of file stencil.h.

Referenced by `make_DXSX_stencil()`.

8.1.3.41 `void mappel::fill_G_stencil ( int size, double stencil[], const double dx[], double theta_sigma ) [inline]`

Definition at line 159 of file stencil.h.

References `square()`.

Referenced by `make_G_stencil()`.

8.1.3.42 `void mappel::fill_gaussian_stencil ( int size, double stencil[], double sigma )`

Definition at line 57 of file stencil.cpp.

References `gauss_norm()`.

Referenced by `make_gaussian_stencil()`.

8.1.3.43 `void mappel::fill_X_stencil ( int size, double stencil[], const double dx[], double theta_sigma ) [inline]`

Definition at line 166 of file stencil.h.

Referenced by `make_X_stencil()`.

8.1.3.44 `double mappel::gamma_prior_grad ( double kappa, double mean, double v ) [inline]`

Definition at line 322 of file stencil.h.

8.1.3.45 `double mappel::gamma_prior_grad2 ( double kappa, double v ) [inline]`

Definition at line 358 of file stencil.h.

8.1.3.46 `double mappel::gauss_norm ( double sigma ) [inline]`

Definition at line 94 of file stencil.h.

Referenced by `fill_gaussian_stencil()`.

**8.1.3.47** `double mappel::gaussian_3D_convolution ( int x, int y, int z, const CubeT & data, const VecFieldT & stencils )`

Definition at line 201 of file stencil.cpp.

Referenced by `estimate_gaussian_3Dmax()`, and `refine_gaussian_3Dmax()`.

**8.1.3.48** `double mappel::gaussian_convolution ( int x, int y, const MatT & data, const VecT & Xstencil, const VecT & Ystencil )`

Definition at line 144 of file stencil.cpp.

Referenced by `estimate_gaussian_2Dmax()`, and `refine_gaussian_2Dmax()`.

**8.1.3.49** `template<class RngT > double mappel::generate_poisson ( RngT & rng, double mu )`

Definition at line 81 of file rng.h.

References `generate_poisson_large()`, and `generate_poisson_small()`.

Referenced by `mappel::methods::simulate_image()`, `simulate_image()`, and `mappel::methods::simulate_image_from←_model()`.

**8.1.3.50** `template<class RngT > IdxT mappel::generate_poisson_large ( RngT & rng, double mu )`

Definition at line 57 of file rng.h.

Referenced by `generate_poisson()`.

**8.1.3.51** `template<class RngT > IdxT mappel::generate_poisson_small ( RngT & rng, double mu )`

Generates a single Poisson distributed int from distribution with mean mu.

#### Parameters

<i>mu</i>	- mean of Poisson distribution
<i>sfmt</i>	- A pointer to the SFMT rng state.

Knuth method circa 1969. Transformed to work in log space. This is linear in mu. Works ok for small counts.

Definition at line 43 of file rng.h.

Referenced by `generate_poisson()`.

**8.1.3.52** `const char * mappel::icontains ( const char * s, const char * pattern )`

Definition at line 45 of file util.cpp.

**8.1.3.53** `bool mappel::is_negative_definite ( const arma::mat & usym )`

Determine if C is negative definite (i.e., -C is positive definite)

## Parameters

<i>usym</i>	A symmetric matrix in upper triangular format.
-------------	------------------------------------------------

## Returns

True if C is negative definite

Definition at line 52 of file numerical.cpp.

References `is_positive_definite()`.

### 8.1.3.54 `bool mappel::is_positive_definite ( const arma::mat & usym )`

Determine if C is positive definite

## Parameters

<i>usym</i>	A symmetric matrix in upper triangular format.
-------------	------------------------------------------------

## Returns

True if C is positive definite

Definition at line 57 of file numerical.cpp.

References `cholesky()`.

Referenced by `is_negative_definite()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

### 8.1.3.55 `bool mappel::is_symmetric ( const arma::mat & A )`

Check that full 2D matrix A is symmetric and can thus be treated as either upper or lower triangular symmetric representation. This will obviously not work with matrices that are already implicitly stored as symmetric triangular format since those matrices won't have the other triangle of elements filled in correctly.

Definition at line 63 of file numerical.cpp.

### 8.1.3.56 `bool mappel::istarts_with ( const char * s, const char * pattern )`

Definition at line 27 of file util.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior()`, `mappel::Gauss2DModel::make_default_prior()`, `mappel::Gauss1DModel::make_default_prior()`, `mappel::Gauss2DsModel::make_default_prior()`, `mappel::methods::make_estimator()`, and `make_estimator()`.

**8.1.3.57** `bool mappel::istarts_with ( const std::string & str, const char * pattern )`

Definition at line 35 of file util.cpp.

**8.1.3.58** `const char* mappel::lambda_term_color ( int size, int Lidx )`

Definition at line 33 of file display.cpp.

References `TERM_BLUE`, `TERM_CYAN`, `TERM_DIM_BLUE`, `TERM_DIM_CYAN`, `TERM_DIM_GREEN`, `TERM_DIM_MAGENTA`, `TERM_DIM_RED`, `TERM_DIM_WHITE`, `TERM_DIM_YELLOW`, `TERM_GREEN`, `TERM_MAGENTA`, `TERM_RED`, `TERM_WHITE`, and `TERM_YELLOW`.

Referenced by `print_image()`.

**8.1.3.59** `double mappel::log_prior_beta2_const ( double beta0, double beta1 ) [inline]`

Definition at line 250 of file stencil.h.

**8.1.3.60** `double mappel::log_prior_beta_const ( double beta ) [inline]`

Definition at line 244 of file stencil.h.

**8.1.3.61** `double mappel::log_prior_gamma_const ( double kappa, double mean ) [inline]`

Definition at line 257 of file stencil.h.

**8.1.3.62** `double mappel::log_prior_normal_const ( double sigma ) [inline]`

Definition at line 269 of file stencil.h.

**8.1.3.63** `double mappel::log_prior_pareto_const ( double alpha, double min ) [inline]`

Definition at line 263 of file stencil.h.

**8.1.3.64** `VecT mappel::make_d_stencil ( int size, double theta_x ) [inline]`

Definition at line 99 of file stencil.h.

References `fill_d_stencil()`.

Referenced by `mappel::Gauss1DsModel::Stencil::Stencil()`, `mappel::Gauss2DModel::Stencil::Stencil()`, `mappel::Gauss1DModel::Stencil::Stencil()`, and `mappel::Gauss2DsModel::Stencil::Stencil()`.

**8.1.3.65** `VecT mappel::make_DX_stencil ( int size, const VecT & Gx, double theta_sigma ) [inline]`

Definition at line 120 of file stencil.h.

References `fill_DX_stencil()`.

Referenced by `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss2DModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, and `mappel::Gauss2DsModel::Stencil::compute_derivatives()`.



**8.1.3.66** `VecT mappel::make_DXS2_stencil ( int size, const VecT & dx, const VecT & Gx, const VecT & DXS, double theta_sigma ) [inline]`

Definition at line 135 of file stencil.h.

References `fill_DXS2_stencil()`.

Referenced by `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, and `mappel::Gauss2DsModel::Stencil::compute_derivatives()`.

**8.1.3.67** `VecT mappel::make_DXS_stencil ( int size, const VecT & dx, const VecT & Gx, double theta_sigma ) [inline]`

Definition at line 127 of file stencil.h.

References `fill_DXS_stencil()`.

Referenced by `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss2DModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, and `mappel::Gauss2DsModel::Stencil::compute_derivatives()`.

**8.1.3.68** `VecT mappel::make_DXSX_stencil ( int size, const VecT & dx, const VecT & Gx, const VecT & DX, double theta_sigma ) [inline]`

Definition at line 143 of file stencil.h.

References `fill_DXSX_stencil()`.

Referenced by `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, and `mappel::Gauss2DsModel::Stencil::compute_derivatives()`.

**8.1.3.69** `VecT mappel::make_G_stencil ( int size, const VecT & dx, double theta_sigma ) [inline]`

Definition at line 106 of file stencil.h.

References `fill_G_stencil()`.

Referenced by `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss2DModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, and `mappel::Gauss2DsModel::Stencil::compute_derivatives()`.

**8.1.3.70** `VecT mappel::make_gaussian_stencil ( int size, double sigma ) [inline]`

Definition at line 218 of file stencil.h.

References `fill_gaussian_stencil()`.

**8.1.3.71** `template<typename T, typename... Args> std::unique_ptr<T> mappel::make_unique ( Args &&... args )`

Definition at line 134 of file util.h.

References `operator<<()`.

**8.1.3.72** `VecT mappel::make_X_stencil ( int size, const VecT & dx, double theta_sigma ) [inline]`

Definition at line 113 of file stencil.h.

References `fill_X_stencil()`.

Referenced by `mappel::Gauss1DsModel::Stencil::Stencil()`, `mappel::Gauss2DModel::Stencil::Stencil()`, `mappel::Gauss1DModel::Stencil::Stencil()`, and `mappel::Gauss2DsModel::Stencil::Stencil()`.

**8.1.3.73** `int mappel::maxidx ( const VecT & v )`

Definition at line 61 of file util.cpp.

**8.1.3.74** `bool mappel::modified_cholesky ( arma::mat & usym )`

#### Parameters

<i>usym</i>	An upper triangular symmetric matrix stored in a full matrix format. This matrix will be overwritten with the upper triangle and diagonal elements of the modified Cholesky decomposition.
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### Returns

true if *usym* was positive semi-definite (no Cholesky modification required). If false we made a modification

Definition at line 128 of file numerical.cpp.

Referenced by `cholesky_make_negative_definite()`, and `cholesky_make_positive_definite()`.

**8.1.3.75** `double mappel::norm_sq ( const VecT & v )`

Definition at line 210 of file numerical.cpp.

References `square()`.

Referenced by `clamp()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**8.1.3.76** `double mappel::normal_prior_grad ( double mu, double sigma )`

**8.1.3.77** `double mappel::normal_prior_grad ( double sigma )`

**8.1.3.78** `double mappel::normal_prior_grad ( double mu, double sigma, double v ) [inline]`

Definition at line 334 of file stencil.h.

**8.1.3.79** `double mappel::normal_prior_grad2 ( double sigma ) [inline]`

Definition at line 370 of file stencil.h.

### 8.1.3.80 `double mappel::normal_quantile_onesided ( double confidence )`

Definition at line 33 of file stencil.cpp.

### 8.1.3.81 `double mappel::normal_quantile_twosided ( double confidence )`

Definition at line 22 of file stencil.cpp.

Referenced by `mappel::methods::error_bounds_expected()`, `mappel::methods::openmp::error_bounds_expected_↵  
stack()`, `mappel::methods::error_bounds_observed()`, and `mappel::methods::openmp::error_bounds_observed_stack()`.

### 8.1.3.82 `std::ostream & mappel::operator<< ( std::ostream & out, const StatsT & stats )`

Definition at line 74 of file util.cpp.

### 8.1.3.83 `template<class Model , typename = EnableIfSubclassT<Model,PointEmitterModel>> std::ostream & mappel::operator<< ( std::ostream & out, const Model & model )`

Definition at line 276 of file PointEmitterModel.h.

### 8.1.3.84 `std::ostream& mappel::operator<< ( std::ostream & out, const Gauss1DModel::Stencil & s )`

Definition at line 164 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::Stencil::derivatives_computed`, `mappel::Gauss1DModel::Stencil::dx`, `mappel::↵  
Gauss1DModel::Stencil::DX`, `mappel::Gauss1DModel::Stencil::DXS`, `mappel::Gauss1DModel::Stencil::Gx`, `print_vec↵  
_row()`, `TERM_BLUE`, `TERM_CYAN`, `TERM_WHITE`, `mappel::Gauss1DModel::Stencil::theta`, and `mappel::Gauss1D↵  
Model::Stencil::X`.

Referenced by `make_unique()`.

### 8.1.3.85 `std::ostream& mappel::operator<< ( std::ostream & out, const Gauss1DsModel::Stencil & s )`

Definition at line 182 of file Gauss1DsModel.cpp.

References `mappel::Gauss1DsModel::Stencil::derivatives_computed`, `mappel::Gauss1DsModel::Stencil::dx`, `mappel↵  
::Gauss1DsModel::Stencil::DX`, `mappel::Gauss1DsModel::Stencil::DXS`, `mappel::Gauss1DsModel::Stencil::DXS2`,  
`mappel::Gauss1DsModel::Stencil::DXSX`, `mappel::Gauss1DsModel::Stencil::Gx`, `print_vec_row()`, `TERM_BLUE`, `TE↵  
RM_CYAN`, `TERM_WHITE`, `mappel::Gauss1DsModel::Stencil::theta`, and `mappel::Gauss1DsModel::Stencil::X`.

### 8.1.3.86 `std::ostream& mappel::operator<< ( std::ostream & out, const Gauss2DModel::Stencil & s )`

Definition at line 249 of file Gauss2DModel.cpp.

References `mappel::Gauss2DModel::Stencil::derivatives_computed`, `mappel::Gauss2DModel::Stencil::dx`, `mappel↵  
Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::dy`, `mappel↵  
Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::Gx`, `mappel↵  
Gauss2DModel::Stencil::Gy`, `print_vec_row()`, `TERM_BLUE`, `TERM_CYAN`, `TERM_WHITE`, `mappel::Gauss2DModel↵  
::Stencil::theta`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

**8.1.3.87** `std::ostream& mappel::operator<< ( std::ostream & out, const Gauss2DsModel::Stencil & s )`

Definition at line 314 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::Stencil::derivatives_computed`, `mappel::Gauss2DsModel::Stencil::dx`, `mappel::Gauss2DsModel::Stencil::DX`, `mappel::Gauss2DsModel::Stencil::DXS`, `mappel::Gauss2DsModel::Stencil::DXS2`, `mappel::Gauss2DsModel::Stencil::DXSX`, `mappel::Gauss2DsModel::Stencil::dy`, `mappel::Gauss2DsModel::Stencil::DY`, `mappel::Gauss2DsModel::Stencil::DYS`, `mappel::Gauss2DsModel::Stencil::DYS2`, `mappel::Gauss2DsModel::Stencil::DYSY`, `mappel::Gauss2DsModel::Stencil::Gx`, `mappel::Gauss2DsModel::Stencil::Gy`, `print_vec_row()`, `TERM_BLUE`, `TERM_CYAN`, `TERM_WHITE`, `mappel::Gauss2DsModel::Stencil::theta`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

**8.1.3.88** `double mappel::pareto_prior_grad ( double alpha, double v ) [inline]`

Definition at line 328 of file stencil.h.

**8.1.3.89** `double mappel::pareto_prior_grad2 ( double alpha, double v ) [inline]`

Definition at line 364 of file stencil.h.

**8.1.3.90** `double mappel::poisson_log_likelihood ( double model_val, double data_val ) [inline]`

Definition at line 226 of file stencil.h.

Referenced by `mappel::methods::likelihood::llh()`, `mappel::methods::likelihood::debug::llh_components()`, and `log_likelihood()`.

**8.1.3.91** `ostream& mappel::print_centered_title ( ostream & out, char fill, int width, const char * title = nullptr )`

Definition at line 83 of file display.cpp.

Referenced by `print_image()`, and `print_labeled_image()`.

**8.1.3.92** `template<class ImageT > std::ostream& mappel::print_image ( std::ostream & out, const ImageT & im )`

**8.1.3.93** `template<> std::ostream& mappel::print_image ( std::ostream & out, const arma::vec & im )`

Definition at line 139 of file display.cpp.

References `print_labeled_image()`.

**8.1.3.94** `template<> std::ostream& mappel::print_image ( std::ostream & out, const arma::mat & im )`

Definition at line 147 of file display.cpp.

References `print_labeled_image()`.

8.1.3.95 `template<> std::ostream& mappel::print_image ( std::ostream & out, const arma::cube & im )`

Definition at line 167 of file display.cpp.

References `lambda_term_color()`, `print_centered_title()`, and `print_labeled_image()`.

8.1.3.96 `ostream& mappel::print_labeled_image ( ostream & out, const arma::mat & im, const char * title, const char * color )`

Definition at line 95 of file display.cpp.

References `print_centered_title()`.

Referenced by `print_image()`, and `print_text_image()`.

8.1.3.97 `template<class ImageT > std::ostream& mappel::print_text_image ( std::ostream & out, const ImageT & im )`

8.1.3.98 `template<> std::ostream& mappel::print_text_image ( std::ostream & out, const arma::vec & im )`

Definition at line 153 of file display.cpp.

References `print_labeled_image()`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

8.1.3.99 `template<> std::ostream& mappel::print_text_image ( std::ostream & out, const arma::mat & im )`

Definition at line 160 of file display.cpp.

References `print_labeled_image()`.

8.1.3.100 `template<class Vec > std::ostream& mappel::print_vec_row ( std::ostream & out, const Vec & vec, const char * header, int header_width, const char * color = nullptr )`

Definition at line 42 of file display.h.

Referenced by `operator<<()`.

8.1.3.101 `void mappel::refine_gaussian_2Dmax ( const MatT & data, const VecT & Xstencil, const VecT & Ystencil, int max_pos[ ] )`

Definition at line 174 of file stencil.cpp.

References `gaussian_convolution()`.

8.1.3.102 `void mappel::refine_gaussian_3Dmax ( const CubeT & data, const VecFieldT & stencils, int max_pos[] )`

Definition at line 242 of file stencil.cpp.

References gaussian\_3D\_convolution().

8.1.3.103 `double mappel::relative_poisson_log_likelihood ( double model_val, double data_val ) [inline]`

Definition at line 235 of file stencil.h.

Referenced by relative\_log\_likelihood(), mappel::methods::likelihood::rllh(), and mappel::methods::likelihood::debug←  
::rllh\_components().

8.1.3.104 `double mappel::restrict_value_range ( double val, double minval, double maxval ) [inline]`

Definition at line 127 of file util.h.

8.1.3.105 `double mappel::rllh_beta2_prior ( double beta0, double beta1, double v, double max = 1., double min = 0. ) [inline]`

Definition at line 282 of file stencil.h.

8.1.3.106 `double mappel::rllh_beta_prior ( double beta, double v, double max = 1., double min = 0. ) [inline]`

Definition at line 275 of file stencil.h.

8.1.3.107 `double mappel::rllh_gamma_prior ( double kappa, double mean, double v ) [inline]`

Definition at line 289 of file stencil.h.

8.1.3.108 `double mappel::rllh_normal_prior ( double mu, double sigma )`

8.1.3.109 `double mappel::rllh_normal_prior ( double mu, double sigma, double v ) [inline]`

Definition at line 301 of file stencil.h.

8.1.3.110 `double mappel::rllh_pareto_prior ( double alpha, double v ) [inline]`

Definition at line 295 of file stencil.h.

8.1.3.111 `template<typename T> int mappel::sgn ( T val )`

sign (signum) function: -1/0/1

Definition at line 120 of file util.h.

Referenced by mappel::estimator::subroutine::compute\_bound\_scaling\_vec().

#### 8.1.3.112 `template<typename T > T mappel::square ( T x )`

Definition at line 125 of file util.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `fill_DXS_stencil()`, `fill_G_stencil()`, `norm_sq()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

### 8.1.4 Variable Documentation

#### 8.1.4.1 `ParallelRngManagerT mappel::rng_manager`

Definition at line 11 of file rng.cpp.

Referenced by `mappel::PointEmitterModel::get_rng_generator()`, `mappel::PointEmitterModel::get_rng_manager()`, `mappel::PointEmitterModel::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, `mappel::PointEmitterModel::sample_prior()`, and `mappel::PointEmitterModel::set_rng_seed()`.

#### 8.1.4.2 `const char * mappel::TERM_BLACK ="1;30"`

Definition at line 13 of file display.cpp.

#### 8.1.4.3 `const char * mappel::TERM_BLUE ="1;34"`

Definition at line 17 of file display.cpp.

Referenced by `lambda_term_color()`, and `operator<<()`.

#### 8.1.4.4 `const char * mappel::TERM_CYAN ="1;36"`

Definition at line 19 of file display.cpp.

Referenced by `lambda_term_color()`, and `operator<<()`.

#### 8.1.4.5 `const char * mappel::TERM_DIM_BLACK ="0;30"`

Definition at line 21 of file display.cpp.

#### 8.1.4.6 `const char * mappel::TERM_DIM_BLUE ="0;34"`

Definition at line 25 of file display.cpp.

Referenced by `lambda_term_color()`.

**8.1.4.7 const char \* mappel::TERM\_DIM\_CYAN ="0;36"**

Definition at line 27 of file display.cpp.

Referenced by lambda\_term\_color().

**8.1.4.8 const char \* mappel::TERM\_DIM\_GREEN ="0;32"**

Definition at line 23 of file display.cpp.

Referenced by lambda\_term\_color().

**8.1.4.9 const char \* mappel::TERM\_DIM\_MAGENTA ="0;35"**

Definition at line 26 of file display.cpp.

Referenced by lambda\_term\_color().

**8.1.4.10 const char \* mappel::TERM\_DIM\_RED ="0;31"**

Definition at line 22 of file display.cpp.

Referenced by lambda\_term\_color().

**8.1.4.11 const char \* mappel::TERM\_DIM\_WHITE ="0;37"**

Definition at line 28 of file display.cpp.

Referenced by lambda\_term\_color().

**8.1.4.12 const char \* mappel::TERM\_DIM\_YELLOW ="0;33"**

Definition at line 24 of file display.cpp.

Referenced by lambda\_term\_color().

**8.1.4.13 const char \* mappel::TERM\_GREEN ="1;32"**

Definition at line 15 of file display.cpp.

Referenced by lambda\_term\_color().

**8.1.4.14 const char \* mappel::TERM\_MAGENTA ="1;35"**

Definition at line 18 of file display.cpp.

Referenced by lambda\_term\_color().



8.1.4.15 `const char * mappel::TERM_RED = "1;31"`

Definition at line 14 of file display.cpp.

Referenced by `lambda_term_color()`.

8.1.4.16 `const char * mappel::TERM_WHITE = "1;37"`

Definition at line 20 of file display.cpp.

Referenced by `lambda_term_color()`, and `operator<<()`.

8.1.4.17 `const char * mappel::TERM_YELLOW = "1;33"`

Definition at line 16 of file display.cpp.

Referenced by `lambda_term_color()`.

## 8.2 mappel::estimator Namespace Reference

### Namespaces

- [subroutine](#)

### Classes

- class [CGaussHeuristicEstimator](#)
- class [CGaussMLE](#)
- class [Estimator](#)
- class [HeuristicEstimator](#)
- class [IterativeMaximizer](#)
- struct [MLEData](#)
- struct [MLEDataStack](#)
- struct [MLEDebugData](#)
- class [NewtonDiagonalMaximizer](#)
- class [NewtonMaximizer](#)
- struct [ProfileBoundsData](#)
- struct [ProfileBoundsDataStack](#)
- struct [ProfileBoundsDebugData](#)
- struct [ProfileLikelihoodData](#)
- class [QuasiNewtonMaximizer](#)
- class [SimulatedAnnealingMaximizer](#)
- class [ThreadedEstimator](#)
- class [TrustRegionMaximizer](#)

## Functions

- `template<class Model >`  
`std::ostream & operator<< (std::ostream &out, Estimator< Model > &estimator)`
- `enum ExitCode : IdxT {`  
`ExitCode::TrustRegionRadius = 9, ExitCode::ModelImprovement = 8, ExitCode::GradRatio = 7, ExitCode::↵`  
`FunctionValue = 6,`  
`ExitCode::StepSize = 5, ExitCode::Success = 4, ExitCode::MaxBacktracks = 3, ExitCode::MaxIter = 2,`  
`ExitCode::Unassigned = 1, ExitCode::Error = 0 }`

### 8.2.1 Class Documentation

#### 8.2.1.1 struct mappel::estimator::MLEData

Data reporting structures A maximum-likelihood estimate for a single image. A container to group the necessary information at an MLEstimate

Definition at line 40 of file estimator.h.

##### Class Members

<a href="#">MatT</a>	obsI	Observed Fisher information matrix at theta.
double	rllh	RLLH at theta.
<a href="#">VecT</a>	theta	Theta estimate.

#### 8.2.1.2 struct mappel::estimator::MLEDataStack

A stack of maximum-likelihood estimates for a stack of images A container to group the necessary information at an MLEstimate

Definition at line 63 of file estimator.h.

##### Class Members

<a href="#">IdxT</a>	Ndata	Number of data estimates.
<a href="#">CubeT</a>	obsI	Observed Fisher information matrix stack. size:[Nparams,Nparams,Ndata].
<a href="#">VecT</a>	rllh	RLLH stack. size:[Ndata].
<a href="#">MatT</a>	theta	Theta estimate stack. size:[Nparams,Ndata].

#### 8.2.1.3 struct mappel::estimator::MLEDebugData

A maximum-likelihood estimate for a single image with debugging information. A container to group the necessary information at an MLEstimate

Definition at line 50 of file estimator.h.

## Class Members

<a href="#">IdxT</a>	Nseq	Number of points evaluated including theta_init and theta_mle.
<a href="#">MatT</a>	obsI	Observed Fisher information matrix at theta.
double	rllh	RLLH at theta.
<a href="#">MatT</a>	sequence	Sequence of evaluated points including theta_init and theta_mle.
<a href="#">VecT</a>	sequence_rllh	RLLH at each point in sequence.
<a href="#">VecT</a>	theta	Theta estimate.

## 8.2.1.4 struct mappel::estimator::ProfileBoundsDebugData

Data for debugging of estimation of profile bounds for a single parameter of a single image Includes both controlling (input) parameters as well as reporting (ouptut) parameters to give output parameters context.

Definition at line 113 of file estimator.h.

## Class Members

<a href="#">IdxT</a>	estimated_idx	Index of single parameter to estimate for.
<a href="#">MLEData</a>	mle	Theta maximum-likelihood estimate, rllh, and ObsI.
<a href="#">IdxT</a>	Nseq_lb	Number of points in sequence_lb.
<a href="#">IdxT</a>	Nseq_ub	Number of points in sequence_ub.
double	profile_lb	size:[Nparams_est] Lower bound estimated for estimated_idx.
double	profile_ub	size:[Nparams_est] Upper bound estimated for estimated_idx.
<a href="#">MatT</a>	sequence_lb	size:[NumParams,Nseq_lb] Sequence of evaluated points for lb estimate (including theta mle as initial point)
<a href="#">VecT</a>	sequence_lb_rllh	size:[Nseq_lb] RLLH at each of the sequence_lb points
<a href="#">MatT</a>	sequence_ub	size:[NumParams,Nseq_ub] Sequence of evaluated points for ub estimate (including theta mle as initial point)
<a href="#">VecT</a>	sequence_ub_rllh	size:[Nseq_ub] RLLH at each of the sequence_ub points
double	target_rllh_delta	Targeted rllh change in value from MLE ( $-\chi^2_{inv}(\text{confidence},1)/2$ )

## 8.2.1.5 struct mappel::estimator::ProfileLikelihoodData

Container for profile liklihood estimator data Includes both controlling (input) parameters as well as reporting (ouptut) parameters to give output parameters context.

Definition at line 74 of file estimator.h.

## Class Members

<a href="#">IdxVecT</a>	fixed_idx	Indexes of fixed parameters.
<a href="#">MatT</a>	fixed_values	Vector values for each fixed parameter size:[Nfixed,Nvalues];.
<a href="#">IdxT</a>	Nfixed	Number of fixed parameters.
<a href="#">IdxT</a>	Nvalues	Number of values of fixed parameters evaluated.
<a href="#">VecT</a>	profile_likelihood	profile likelihood for each column of fixed parameter values
<a href="#">MatT</a>	profile_parameters	Points at which the profile likelihood maximum was obtained.

## 8.2.2 Enumeration Type Documentation

### 8.2.2.1 `enum mappel::estimator::ExitCode : IdxT` [strong]

Enumerated exit codes for estimation methods

- Error: A Numerical Error was caught. Did not converge.
- Unassigned: Logical error if this is still set
- MaxIter: Max iterations exceeded. Did not converge.
- MaxBacktracks: Backtracking failed. Did not converge successfully.
- Success: Successful completion
- StepSize: Relative Step size was less than epsilon. Converged successfully.
- FunctionValue: Function value change was less than epsilon. Converged successfully.
- GradRatio: Grad ratio was less than epsilon. Converged successfully.
- ModellImprovement: Model predicted improvement is less than epsilon. Converged Successfully
- TrustRegionRadius: Trust region size was less than epsilon. Converged successfully.

Enumerator

***TrustRegionRadius***

***ModellImprovement***

***GradRatio***

***FunctionValue***

***StepSize***

***Success***

***MaxBacktracks***

***MaxIter***

***Unassigned***

***Error***

Definition at line 172 of file estimator.h.

## 8.2.3 Function Documentation

### 8.2.3.1 `template<class Model > std::ostream& mappel::estimator::operator<< ( std::ostream & out, Estimator< Model > & estimator )`

Definition at line 351 of file estimator\_impl.h.

## 8.3 mappel::estimator::subroutine Namespace Reference

### Functions

- [VecT solve\\_profile\\_initial\\_step](#) (const [MatT](#) &obsI, [IdxT](#) fixed\_idx, double llh\_delta)
- [VecT bound\\_step](#) (const [VecT](#) &step, const [VecT](#) &theta, const [VecT](#) &lbound, const [VecT](#) &ubound)
- void [compute\\_bound\\_scaling\\_vec](#) (const [VecT](#) &theta, const [VecT](#) &g, const [VecT](#) &lbound, const [VecT](#) &ubound, [VecT](#) &v, [VecT](#) &Jv)
- [VecT compute\\_D\\_scale](#) (const [VecT](#) &oldDscale, const [VecT](#) &grad2)
- void [compute\\_scaled\\_problem](#) (const [MatT](#) &H, const [VecT](#) &g, const [VecT](#) &Dinv, const [VecT](#) &Jv, [MatT](#) &Hhat, [VecT](#) &ghat)
- double [compute\\_initial\\_trust\\_radius](#) (const [VecT](#) &ghat)
- [VecT compute\\_cauchy\\_point](#) (const [VecT](#) &g, const [MatT](#) &H, double delta)
- double [compute\\_quadratic\\_model\\_value](#) (const [VecT](#) &s, const [VecT](#) &g, const [MatT](#) &H)
- *Quadratic model value at given step Compute a quadratic model.*
- [VecT solve\\_TR\\_subproblem](#) (const [VecT](#) &g, const [MatT](#) &H, double delta)
- *Exact solver the TR sub-problem even for non-positive definite H.*
- [VecT solve\\_restricted\\_step\\_length\\_newton](#) (const [VecT](#) &g, const [MatT](#) &H, double delta, double lambda\_lb, double lambda\_ub)

### 8.3.1 Detailed Description

Estimation subroutines common to several estimators and independent of the Model

Common subroutines shared between estimators.

These methods are model agnostic.

### 8.3.2 Function Documentation

**8.3.2.1** [VecT mappel::estimator::subroutine::bound\\_step](#) ( const [VecT](#) & *step*, const [VecT](#) & *theta*, const [VecT](#) & *lbound*, const [VecT](#) & *ubound* )

Return a new step that is guaranteed to keep theta in the interior of the feasible region. Uses a relative backtracking technique to step away from the boundary into the interior.

#### Parameters

<i>step</i>	proposed step
<i>theta</i>	current theta
<i>lbound</i>	lower bounds
<i>ubound</i>	upper bounds

#### Returns

bounded step

Definition at line 82 of file estimator.cpp.

Referenced by mappel::estimator::IterativeMaximizer< Model >::local\_profile\_maximize().

**8.3.2.2** void mappel::estimator::subroutine::compute\_bound\_scaling\_vec ( const VecT & *theta*, const VecT & *g*, const VecT & *lbound*, const VecT & *ubound*, VecT & *v*, VecT & *Jv* )

Bounds scaling vector for affine scaling of bounds constrained optimization problems. This *v* is from Coleman&Li (1996). It represents a scaling factor for bound constrained problems. For unconstrained problems  $v = \text{sgn}(\text{grad})$ ;

#### Parameters

in	<i>theta</i>	current theta
in	<i>g</i>	gradient
in	<i>lbound</i>	lower bound
in	<i>ubound</i>	upper bound
out	<i>v</i>	Scaling vector
out	<i>Jv</i>	Jacobian

Definition at line 132 of file estimator.cpp.

References mappel::sgn().

Referenced by mappel::estimator::IterativeMaximizer< Model >::local\_profile\_maximize().

**8.3.2.3** VecT mappel::estimator::subroutine::compute\_cauchy\_point ( const VecT & *g*, const MatT & *H*, double *delta* )

Definition at line 175 of file estimator.cpp.

Referenced by mappel::estimator::IterativeMaximizer< Model >::local\_profile\_maximize().

**8.3.2.4** VecT mappel::estimator::subroutine::compute\_D\_scale ( const VecT & *oldDscale*, const VecT & *grad2* )

Compute an affine scaling diagonal matrix to scale problem away from boundaries. This works for either minimization or maximization.  $\text{sign}(\text{grad2})$  is not important

#### Parameters

<i>oldDscale</i>	Last D scaling matrix
<i>grad2</i>	Diagonal of hessian matrix

#### Returns

Diagonal scaling matrix as a vector.

Definition at line 159 of file estimator.cpp.

References mappel::clamp().

#### 8.3.2.5 double mappel::estimator::subroutine::compute\_initial\_trust\_radius ( const VecT & *ghat* )

Definition at line 170 of file estimator.cpp.

References mappel::clamp().

Referenced by mappel::estimator::IterativeMaximizer< Model >::local\_profile\_maximize().

#### 8.3.2.6 double mappel::estimator::subroutine::compute\_quadratic\_model\_value ( const VecT & *s*, const VecT & *g*, const MatT & *H* )

Quadratic model value at given step Compute a quadratic model.

Definition at line 183 of file estimator.cpp.

Referenced by mappel::estimator::IterativeMaximizer< Model >::local\_profile\_maximize().

#### 8.3.2.7 void mappel::estimator::subroutine::compute\_scaled\_problem ( const MatT & *H*, const VecT & *g*, const VecT & *Dinv*, const VecT & *Jv*, MatT & *Hhat*, VecT & *ghat* )

Definition at line 164 of file estimator.cpp.

Referenced by mappel::estimator::IterativeMaximizer< Model >::local\_profile\_maximize().

#### 8.3.2.8 VecT mappel::estimator::subroutine::solve\_profile\_initial\_step ( const MatT & *obsI*, IdxT *fixed\_idx*, double *llh\_delta* )

Find initial step lengths in profile bounds estimation VM algorithm

Definition at line 52 of file estimator.cpp.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound\_debug(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_parallel(), and mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

#### 8.3.2.9 VecT mappel::estimator::subroutine::solve\_restricted\_step\_length\_newton ( const VecT & *g*, const MatT & *H*, double *delta*, double *lambda\_lb*, double *lambda\_ub* )

Definition at line 256 of file estimator.cpp.

References mappel::cholesky(), mappel::cholesky\_convert\_lower\_triangular(), mappel::cholesky\_solve(), and mappel::clamp().

Referenced by solve\_TR\_subproblem().

### 8.3.2.10 VecT mappel::estimator::subroutine::solve\_TR\_subproblem ( const VecT & g, const MatT & H, double delta )

Exact solver the TR sub-problem even for non-positive definite H.

This method is a hybrid technique mixing ideas from Geyer (2013) and the "trust" R-package Nocedal and Wright (2000) More and Sorensen (1981)

Definition at line 189 of file estimator.cpp.

References mappel::cholesky(), mappel::cholesky\_solve(), and solve\_restricted\_step\_length\_newton().

Referenced by mappel::estimator::IterativeMaximizer< Model >::local\_profile\_maximize().

## 8.4 mappel::mcmc Namespace Reference

### Classes

- struct [MCMCData](#)
- struct [MCMCDataStack](#)
- struct [MCMCDebugData](#)

### Functions

- [IdxT num\\_oversample](#) (IdxT Nsample, IdxT Nburnin, IdxT thin)
- [MatT thin\\_sample](#) (MatT &sample, IdxT Nburnin, IdxT thin)
- void [thin\\_sample](#) (const MatT &sample, const VecT &sample\_rllh, IdxT Nburnin, IdxT thin, MatT &subsample, VecT &subsample\_rllh)
- void [estimate\\_sample\\_posterior](#) (const MatT &sample, VecT &theta\_posterior\_mean, MatT &theta\_posterior\_cov)
- template<class Mat, class Vec>  
void [compute\\_posterior\\_credible](#) (const Mat &sample, double confidence, Vec &lb, Vec &ub)
- template<class Model>  
void [sample\\_posterior](#) (const Model &model, const ModelDataT< Model > &im, const StencilT< Model > &theta\_init, MatT &sample, VecT &sample\_rllh)
- template<class Model>  
void [sample\\_posterior\\_debug](#) (const Model &model, const ModelDataT< Model > &im, const StencilT< Model > &theta\_init, MatT &sample, VecT &sample\_rllh, MatT &candidate, VecT &candidate\_rllh)

### 8.4.1 Function Documentation

#### 8.4.1.1 template<class Mat, class Vec> void mappel::mcmc::compute\_posterior\_credible ( const Mat & sample, double confidence, Vec & lb, Vec & ub )

Definition at line 32 of file mcmc.h.

Referenced by mappel::methods::error\_bounds\_posterior\_credible(), and mappel::methods::estimate\_posterior().



8.4.1.2 `void mappel::mcmc::estimate_sample_posterior ( const MatT & sample, VecT & theta_posterior_mean, MatT & theta_posterior_cov ) [inline]`

Definition at line 25 of file `mcmc.h`.

Referenced by `mappel::methods::estimate_posterior()`.

8.4.1.3 `IdxT mappel::mcmc::num_oversample ( IdxT Nsample, IdxT Nburnin, IdxT thin )`

Definition at line 40 of file `mcmc.cpp`.

Referenced by `mappel::methods::estimate_posterior()`, and `mappel::methods::openmp::estimate_posterior_stack()`.

8.4.1.4 `template<class Model > void mappel::mcmc::sample_posterior ( const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & theta_init, MatT & sample, VecT & sample_rllh )`

Definition at line 41 of file `mcmc.h`.

References `mappel::methods::objective::rllh()`.

Referenced by `mappel::methods::estimate_posterior()`.

8.4.1.5 `template<class Model > void mappel::mcmc::sample_posterior_debug ( const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & theta_init, MatT & sample, VecT & sample_rllh, MatT & candidate, VecT & candidate_rllh )`

Definition at line 73 of file `mcmc.h`.

References `mappel::methods::objective::rllh()`.

Referenced by `mappel::methods::debug::estimate_posterior_debug()`.

8.4.1.6 `MatT mappel::mcmc::thin_sample ( MatT & sample, IdxT Nburnin, IdxT thin )`

Definition at line 46 of file `mcmc.cpp`.

References `mappel::mcmc::MCMCData::thin`.

Referenced by `mappel::methods::estimate_posterior()`.

8.4.1.7 `void mappel::mcmc::thin_sample ( const MatT & sample, const VecT & sample_rllh, IdxT Nburnin, IdxT thin, MatT & subsample, VecT & subsample_rllh )`

Definition at line 57 of file `mcmc.cpp`.

References `mappel::mcmc::MCMCData::sample_rllh`, and `mappel::mcmc::MCMCData::thin`.

## 8.5 mappel::methods Namespace Reference

Templated functions for operating on a [PointEmitterModel](#).

## Namespaces

- [debug](#)
- [likelihood](#)
- [objective](#)
- [openmp](#)

## Functions

- `template<class Model >`  
[ReturnIfSubclassT](#)< [ImageT](#)< Model >, Model, [ImageFormat1DBase](#) > [model\\_image](#) (const Model &model, const [StencilT](#)< Model > &s)
- `template<class Model >`  
[ReturnIfSubclassT](#)< [ImageT](#)< Model >, Model, [ImageFormat2DBase](#) > [model\\_image](#) (const Model &model, const typename Model::Stencil &s)
- `template<class Model >`  
[ImageT](#)< Model > [model\\_image](#) (const Model &model, const [ParamT](#)< Model > &theta)
- `template<class Model , class rng_t >`  
[ModelDataT](#)< Model > [simulate\\_image](#) (const Model &model, const [ParamT](#)< Model > &theta)
- `template<class Model , class rng_t >`  
[ModelDataT](#)< Model > [simulate\\_image](#) (const Model &model, const [ParamT](#)< Model > &theta, rng\_t &rng)
- `template<class Model >`  
[ModelDataT](#)< Model > [simulate\\_image](#) (const Model &model, const [StencilT](#)< Model > &s)
- `template<class Model >`  
[ModelDataT](#)< Model > [simulate\\_image\\_from\\_model](#) (const Model &model, const [ImageT](#)< Model > &model←\_im)
- `template<class Model >`  
void [aposteriori\\_objective](#) (const Model &model, const [ModelDataT](#)< Model > &data\_im, const [StencilT](#)< Model > &s, double &rllh, [ParamT](#)< Model > &grad, [MatT](#) &hess)
- `template<class Model >`  
void [aposteriori\\_objective](#) (const Model &model, const [ModelDataT](#)< Model > &data\_im, const [ParamT](#)< Model > &theta, double &rllh, [ParamT](#)< Model > &grad, [MatT](#) &hess)
- `template<class Model >`  
void [prior\\_objective](#) (const Model &model, const [ParamT](#)< Model > &theta, double &rllh, [ParamT](#)< Model > &grad, [MatT](#) &hess)
- `template<class Model >`  
void [likelihood\\_objective](#) (const Model &model, const [ModelDataT](#)< Model > &data\_im, const [StencilT](#)< Model > &s, double &rllh, [ParamT](#)< Model > &grad, [MatT](#) &hess)
- `template<class Model >`  
void [likelihood\\_objective](#) (const Model &model, const [ModelDataT](#)< Model > &data\_im, const [ParamT](#)< Model > &theta, double &rllh, [ParamT](#)< Model > &grad, [MatT](#) &hess)
- `template<class Model >`  
[ParamT](#)< Model > [cr\\_lower\\_bound](#) (const Model &model, const typename Model::Stencil &s)  
*Calculate the Cramer-Rao lower bound at the given parameters.*
- `template<class Model >`  
[ParamT](#)< Model > [cr\\_lower\\_bound](#) (const Model &model, const [ParamT](#)< Model > &theta)
- `template<class Model >`  
[MatT](#) [expected\\_information](#) (const Model &model, const [ParamT](#)< Model > &theta)
- `template<class Model >`  
[MatT](#) [observed\\_information](#) (const Model &model, const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_mle)

- template<class Model >  
MatT [observed\\_information](#) (const Model &model, const [ModelDataT](#)< Model > &data, const [StencilT](#)< Model > &theta\_mle)
- template<class Model >  
void [estimate\\_max](#) (const Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, [estimator::MLEData](#) &mle)
- template<class Model >  
void [estimate\\_max](#) (const Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, const [ParamT](#)< Model > &theta\_init, [estimator::MLEData](#) &mle)
- template<class Model >  
void [estimate\\_max](#) (const Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, [estimator::MLEData](#) &mle, [StatsT](#) &stats)
- template<class Model >  
void [estimate\\_max](#) (const Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, const [ParamT](#)< Model > &theta\_init, [estimator::MLEData](#) &mle, [StatsT](#) &stats)
- template<class Model >  
double [estimate\\_profile\\_likelihood](#) (const Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, const [IdxVecT](#) &fixed\_idx, const [ParamT](#)< Model > &fixed\_theta\_init)
- template<class Model >  
double [estimate\\_profile\\_likelihood](#) (const Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, const [IdxVecT](#) &fixed\_idx, const [ParamT](#)< Model > &fixed\_theta\_init, [StencilT](#)< Model > &profile←\_max)
- template<class Model >  
double [estimate\\_profile\\_likelihood](#) (const Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, const [IdxVecT](#) &fixed\_idx, const [ParamT](#)< Model > &fixed\_theta\_init, [StencilT](#)< Model > &profile←\_max, [StatsT](#) &stats)
- template<class Model >  
void [estimate\\_profile\\_likelihood](#) (const Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, const [ParamT](#)< Model > &theta\_init, [estimator::ProfileLikelihoodData](#) &profile\_data)
- template<class Model >  
void [estimate\\_posterior](#) (const Model &model, const [ModelDataT](#)< Model > &data, [mcmc::MCMCData](#) &mcmc←\_est)
- template<class Model >  
void [estimate\\_posterior](#) (const Model &model, const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [mcmc::MCMCData](#) &mcmc\_est)
- template<class Model >  
void [error\\_bounds\\_expected](#) (const Model &model, const [ParamT](#)< Model > &theta\_est, double confidence, [ParamT](#)< Model > &theta\_lb, [ParamT](#)< Model > &theta\_ub)
- template<class Model >  
void [error\\_bounds\\_observed](#) (const Model &model, const [estimator::MLEData](#) &mle, double confidence, [ParamT](#)< Model > &theta\_lb, [ParamT](#)< Model > &theta\_ub)
- template<class Model >  
void [error\\_bounds\\_profile\\_likelihood](#) (const Model &model, const [ModelDataT](#)< Model > &data, [estimator::←ProfileBoundsData](#) &bounds)
- template<class Model >  
void [error\\_bounds\\_profile\\_likelihood](#) (const Model &model, const [ModelDataT](#)< Model > &data, [estimator::←ProfileBoundsData](#) &bounds, [StatsT](#) &stats)
- template<class Model >  
void [error\\_bounds\\_posterior\\_credible](#) (const Model &model, const MatT &sample, double confidence, [ParamT](#)< Model > &theta\_lb, [ParamT](#)< Model > &theta\_ub)
- template<class Model >  
Model::ImageT [model\\_image](#) (const Model &model, const [ParamT](#)< Model > &theta)

- `template<class Model >`  
`ModelDataT< Model > simulate_image (const Model &model, const ParamT< Model > &theta)`
- `template<class Model , class RngT >`  
`ModelDataT< Model > simulate_image (const Model &model, const ParamT< Model > &theta, RngT &rng)`
- `template<class Model , class rng_t >`  
`ReturnIfSubclassT< ModelDataT< Model >, Model, PoissonNoise1DObjective > simulate_image (const Model &model, const StencilT< Model > &s, rng_t &rng)`  
*Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise1DObjective](#).*
- `template<class Model , class rng_t >`  
`ReturnIfSubclassT< ModelDataT< Model >, Model, PoissonNoise1DObjective > simulate_image_from_model (const Model &model, const ImageT< Model > &model_im, rng_t &rng)`  
*Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise1DObjective](#).*
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, PoissonNoise1DObjective > expected_information (const Model &model, const StencilT< Model > &s)`  
*Compute the expected information (Fisher information at theta). Note: Expected information is an average quantity and is independent of the data. Enabled for [PoissonNoise1DObjective](#).*
- `template<class Model >`  
`ReturnIfSubclassT< std::unique_ptr< estimator::Estimator< Model > >, Model, PoissonNoise1DObjective > make_estimator (Model &model, std::string ename)`
- `template<class Model , class rng_t >`  
`ReturnIfSubclassT< ImageT< Model >, Model, PoissonNoise2DObjective > simulate_image (const Model &model, const StencilT< Model > &s, rng_t &rng)`  
*Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise2DObjective](#).*
- `template<class Model , class rng_t >`  
`ReturnIfSubclassT< ImageT< Model >, Model, PoissonNoise2DObjective > simulate_image_from_model (const Model &model, const ImageT< Model > &model_im, rng_t &rng)`  
*Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise2DObjective](#).*
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, PoissonNoise2DObjective > expected_information (const Model &model, const StencilT< Model > &s)`  
*Compute the expected information (Fisher information at theta). Note: Expected information is an average quantity and is independent of the data. Enabled for [PoissonNoise2DObjective](#).*
- `template<class Model >`  
`ReturnIfSubclassT< std::unique_ptr< estimator::Estimator< Model > >, Model, PoissonNoise2DObjective > make_estimator (Model &model, std::string ename)`

### 8.5.1 Detailed Description

Templated functions for operating on a [PointEmitterModel](#).

Most methods are overloaded to take a `ParamT` or a `StencilT`. The precomputed stencil for a `theta` value contains the common computational values needed by all methods that compute the likelihood function or its derivatives. Note that methods in `model::prior::` namespace do not take a stencil (or data) as they are independent of the data and the likelihood function.

Methods with `xxx_components` return a sequence of values representing the results from each pixel in turn. The sum of these components is the overall model value. (e.g. `sum(llh_components(...)) == llh(...)`). These methods are useful for detailed inspection of the contributions of each pixel or prior component to the overall result. External template based methods for [PointEmitterModel](#)'s. These are general or convenience functions that are included in this file. Those methods specific to other sub-types of Models should be included within that sub-type's .h file, using the `enable_if` mechanism to restrict their instantiation to the correct sub-types.

## 8.5.2 Function Documentation

**8.5.2.1** `template<class Model > void mappel::methods::aposteriori_objective ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s, double & rlh, ParamT< Model > & grad, MatT & hess )`

Definition at line 218 of file `model_methods_impl.h`.

References `mappel::methods::likelihood::hessian()`, and `mappel::methods::likelihood::rlh()`.

Referenced by `aposteriori_objective()`.

**8.5.2.2** `template<class Model > void mappel::methods::aposteriori_objective ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta, double & rlh, ParamT< Model > & grad, MatT & hess )`

Definition at line 256 of file `model_methods_impl.h`.

References `aposteriori_objective()`, `mappel::methods::objective::grad()`, and `mappel::methods::objective::rlh()`.

**8.5.2.3** `template<class Model > ParamT< Model > mappel::methods::cr_lower_bound ( const Model & model, const typename Model::Stencil & s )`

Calculate the Cramer-Rao lower bound at the given parameters.

## Parameters

in	<i>theta</i>	The parameters to evaluate the CRLB at
out	<i>crlb</i>	The calculated parameters

Definition at line 283 of file `model_methods_impl.h`.

References `expected_information()`.

Referenced by `cr_lower_bound()`, and `error_bounds_expected()`.

**8.5.2.4** `template<class Model > ParamT< Model > mappel::methods::cr_lower_bound ( const Model & model, const ParamT< Model > & theta )`

Definition at line 295 of file `model_methods_impl.h`.

References `cr_lower_bound()`.

**8.5.2.5** `template<class Model > void mappel::methods::error_bounds_expected ( const Model & model, const ParamT< Model > & theta_est, double confidence, ParamT< Model > & theta_lb, ParamT< Model > & theta_ub )`

Definition at line 405 of file `model_methods_impl.h`.

References `cr_lower_bound()`, and `mappel::normal_quantile_twosided()`.

**8.5.2.6** `template<class Model > void mappel::methods::error_bounds_observed ( const Model & model, const estimator::MLEData & mle, double confidence, ParamT< Model > & theta_lb, ParamT< Model > & theta_ub )`

Definition at line 416 of file `model_methods_impl.h`.

References `mappel::normal_quantile_twosided()`, `mappel::estimator::MLEData::obsI`, and `mappel::estimator::MLEData::theta`.

**8.5.2.7** `template<class Model > void mappel::methods::error_bounds_posterior_credible ( const Model & model, const MatT & sample, double confidence, ParamT< Model > & theta_lb, ParamT< Model > & theta_ub )`

Definition at line 444 of file `model_methods_impl.h`.

References `mappel::mcmc::compute_posterior_credible()`.

**8.5.2.8** `template<class Model > void mappel::methods::error_bounds_profile_likelihood ( const Model & model, const ModelDataT< Model > & data, estimator::ProfileBoundsData & bounds )`

Definition at line 426 of file `model_methods_impl.h`.

References `mappel::chisq_quantile()`, `mappel::estimator::ProfileBoundsData::confidence`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, and `mappel::estimator::ProfileBoundsData::target_rllh_delta`.

**8.5.2.9** `template<class Model > void mappel::methods::error_bounds_profile_likelihood ( const Model & model, const ModelDataT< Model > & data, estimator::ProfileBoundsData & bounds, StatsT & stats )`

Definition at line 435 of file `model_methods_impl.h`.

References `mappel::chisq_quantile()`, `mappel::estimator::ProfileBoundsData::confidence`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::iterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::ProfileBoundsData::target_rllh_delta`.

**8.5.2.10** `template<class Model > void mappel::methods::estimate_max ( const Model & model, const ModelDataT< Model > & data, const std::string & method, estimator::MLEData & mle )`

Definition at line 322 of file `model_methods_impl.h`.

References `make_estimator()`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, and `mappel::estimator::Estimator< Model >::~Estimator()`.

**8.5.2.11** `template<class Model > void mappel::methods::estimate_max ( const Model & model, const ModelDataT< Model > & data, const std::string & method, const ParamT< Model > & theta_init, estimator::MLEData & mle )`

Definition at line 331 of file `model_methods_impl.h`.

References `make_estimator()`.

8.5.2.12 `template<class Model > void mappel::methods::estimate_max ( const Model & model, const ModelDataT< Model > & data, const std::string & method, estimator::MLEData & mle, StatsT & stats )`

Definition at line 338 of file `model_methods_impl.h`.

References `make_estimator()`.

8.5.2.13 `template<class Model > void mappel::methods::estimate_max ( const Model & model, const ModelDataT< Model > & data, const std::string & method, const ParamT< Model > & theta_init, estimator::MLEData & mle, StatsT & stats )`

Definition at line 348 of file `model_methods_impl.h`.

References `make_estimator()`.

8.5.2.14 `template<class Model > void mappel::methods::estimate_posterior ( const Model & model, const ModelDataT< Model > & data, mcmc::MCMCData & mcmc_est )`

Definition at line 383 of file `model_methods_impl.h`.

8.5.2.15 `template<class Model > void mappel::methods::estimate_posterior ( const Model & model, const ModelDataT< Model > & data, const ParamT< Model > & theta_init, mcmc::MCMCData & mcmc_est )`

Definition at line 391 of file `model_methods_impl.h`.

References `mappel::mcmc::compute_posterior_credible()`, `mappel::mcmc::MCMCData::confidence`, `mappel::mcmc::MCMCData::credible_lb`, `mappel::mcmc::MCMCData::credible_ub`, `mappel::mcmc::estimate_sample_posterior()`, `mappel::mcmc::MCMCData::Nburnin`, `mappel::mcmc::MCMCData::Nsample`, `mappel::mcmc::num_oversample()`, `mappel::mcmc::MCMCData::sample`, `mappel::mcmc::MCMCData::sample_cov`, `mappel::mcmc::MCMCData::sample_mean`, `mappel::mcmc::sample_posterior()`, `mappel::mcmc::MCMCData::sample_rllh`, `mappel::mcmc::MCMCData::thin`, and `mappel::mcmc::thin_sample()`.

8.5.2.16 `template<class Model > double mappel::methods::estimate_profile_likelihood ( const Model & model, const ModelDataT< Model > & data, const std::string & method, const IdxVecT & fixed_idxs, const ParamT< Model > & fixed_theta_init )`

Definition at line 357 of file `model_methods_impl.h`.

References `make_estimator()`.

8.5.2.17 `template<class Model > double mappel::methods::estimate_profile_likelihood ( const Model & model, const ModelDataT< Model > & data, const std::string & method, const IdxVecT & fixed_idxs, const ParamT< Model > & fixed_theta_init, StencilT< Model > & profile_max )`

Definition at line 365 of file `model_methods_impl.h`.

References `make_estimator()`.

8.5.2.18 `template<class Model > double mappel::methods::estimate_profile_likelihood ( const Model & model, const ModelDataT< Model > & data, const std::string & method, const IdxVecT & fixed_idxs, const ParamT< Model > & fixed_theta_init, StencilT< Model > & profile_max, StatsT & stats )`

Definition at line 373 of file `model_methods_impl.h`.

References `make_estimator()`.

8.5.2.19 `template<class Model > void mappel::methods::estimate_profile_likelihood ( const Model & model, const ModelDataT< Model > & data, const std::string & method, const ParamT< Model > & theta_init, estimator::ProfileLikelihoodData & profile_data )`

8.5.2.20 `template<class Model > ReturnIfSubclassT<MatT, Model, PoissonNoise1DObjective>  
mappel::methods::expected_information ( const Model & model, const StencilT< Model > & s )`

Compute the expected information (Fisher information at  $\theta$ ). Note: Expected information is an average quantity and is independent of the data. Enabled for [PoissonNoise1DObjective](#).

#### Parameters

<i>model</i>	<a href="#">PointEmitterModel</a>
<i>s</i>	Stencil at desired $\theta$

#### Returns

The fisher information matrix as an symmetric matrix in upper-triangular format

Definition at line 77 of file `PoissonNoise1DObjective.h`.

8.5.2.21 `template<class Model > ReturnIfSubclassT<MatT, Model, PoissonNoise2DObjective>  
mappel::methods::expected_information ( const Model & model, const StencilT< Model > & s )`

Compute the expected information (Fisher information at  $\theta$ ). Note: Expected information is an average quantity and is independent of the data. Enabled for [PoissonNoise2DObjective](#).

#### Parameters

<i>model</i>	<a href="#">PolImageCoordTEmitterModel</a>
<i>s</i>	Stencil at desired $\theta$

#### Returns

The fisher information matrix as an symmetric matrix in upper-triangular format

Definition at line 83 of file `PoissonNoise2DObjective.h`.

References `mappel::ImageFormat2DBase::size`.



**8.5.2.22** `template<class Model> MatT mappel::methods::expected_information ( const Model & model, const ParamT< Model> & theta )`

Definition at line 301 of file `model_methods_impl.h`.

Referenced by `cr_lower_bound()`.

**8.5.2.23** `template<class Model> void mappel::methods::likelihood_objective ( const Model & model, const ModelDataT< Model> & data_im, const StencilT< Model> & s, double & rllh, ParamT< Model> & grad, MatT & hess )`

Definition at line 247 of file `model_methods_impl.h`.

References `mappel::methods::likelihood::hessian()`, and `mappel::methods::likelihood::rllh()`.

Referenced by `likelihood_objective()`.

**8.5.2.24** `template<class Model> void mappel::methods::likelihood_objective ( const Model & model, const ModelDataT< Model> & data_im, const ParamT< Model> & theta, double & rllh, ParamT< Model> & grad, MatT & hess )`

Definition at line 270 of file `model_methods_impl.h`.

References `mappel::methods::objective::grad()`, `likelihood_objective()`, and `mappel::methods::objective::rllh()`.

**8.5.2.25** `template<class Model> ReturnIfSubclassT<std::unique_ptr<estimator::Estimator<Model>>, Model, PoissonNoise1DObjective> mappel::methods::make_estimator ( Model & model, std::string ename )`

Definition at line 95 of file `PoissonNoise1DObjective.h`.

References `mappel::istarts_with()`.

Referenced by `estimate_max()`, `mappel::methods::debug::estimate_max_debug()`, `mappel::methods::openmp::estimate_max_stack()`, `estimate_profile_likelihood()`, and `mappel::methods::openmp::estimate_profile_likelihood_stack()`.

**8.5.2.26** `template<class Model> ReturnIfSubclassT<std::unique_ptr<estimator::Estimator<Model>>, Model, PoissonNoise2DObjective> mappel::methods::make_estimator ( Model & model, std::string ename )`

Definition at line 100 of file `PoissonNoise2DObjective.h`.

References `mappel::istarts_with()`.

**8.5.2.27** `template<class Model> Model::ImageT mappel::methods::model_image ( const Model & model, const ParamT< Model> & theta )`

Expected number of photons at each pixel in image given the emitter model

Definition at line 16 of file `model_methods_impl.h`.

References `model_image()`.

**8.5.2.28** `template<class Model > ImageT<Model> mappel::methods::model_image ( const Model & model, const ParamT<Model > & theta )`

Expected number of photons at each pixel in image given the emitter model

Definition at line 16 of file `model_methods_impl.h`.

References `model_image()`.

**8.5.2.29** `template<class Model > ReturnIfSubclassT<ImageT<Model>, Model, ImageFormat2DBase> mappel::methods::model_image ( const Model & model, const typename Model::Stencil & s )`

Definition at line 122 of file `ImageFormat2DBase.h`.

References `mappel::ImageFormat2DBase::size`.

**8.5.2.30** `template<class Model > ReturnIfSubclassT<ImageT<Model>, Model, ImageFormat1DBase> mappel::methods::model_image ( const Model & model, const StencilT<Model > & s )`

Definition at line 125 of file `ImageFormat1DBase.h`.

Referenced by `model_image()`, and `mappel::methods::openmp::simulate_image_stack()`.

**8.5.2.31** `template<class Model > MatT mappel::methods::observed_information ( const Model & model, const ModelDataT<Model > & data, const ParamT<Model > & theta_mle )`

Definition at line 315 of file `model_methods_impl.h`.

Referenced by `mappel::estimator::IterativeMaximizer<Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer<Model >::compute_estimate_debug()`, `mappel::estimator::HeuristicEstimator<Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE<Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer<Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer<Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator<Model >::record_exit_code()`.

**8.5.2.32** `template<class Model > MatT mappel::methods::observed_information ( const Model & model, const ModelDataT<Model > & data, const StencilT<Model > & theta_mle )`

Definition at line 307 of file `model_methods_impl.h`.

References `mappel::methods::objective::hessian()`.

**8.5.2.33** `template<class Model > void mappel::methods::prior_objective ( const Model & model, const ParamT<Model > & theta, double & rlh, ParamT<Model > & grad, MatT & hess )`

Definition at line 229 of file `model_methods_impl.h`.

8.5.2.34 `template<class Model > ModelDataT<Model> mappel::methods::simulate_image ( const Model & model, const ParamT< Model > & theta )`

Definition at line 22 of file `model_methods_impl.h`.

References `simulate_image()`.

Referenced by `simulate_image()`, and `mappel::methods::openmp::simulate_image_stack()`.

8.5.2.35 `template<class Model , class RngT > ModelDataT<Model> mappel::methods::simulate_image ( const Model & model, const ParamT< Model > & theta, RngT & rng )`

Definition at line 29 of file `model_methods_impl.h`.

References `simulate_image()`.

8.5.2.36 `template<class Model , class rng_t > ModelDataT<Model> mappel::methods::simulate_image ( const Model & model, const ParamT< Model > & theta )`

Definition at line 22 of file `model_methods_impl.h`.

References `simulate_image()`.

Referenced by `simulate_image()`, and `mappel::methods::openmp::simulate_image_stack()`.

8.5.2.37 `template<class Model , class rng_t > ReturnIfSubclassT<ModelDataT<Model>, Model, PoissonNoise1DObjective> mappel::methods::simulate_image ( const Model & model, const StencilT< Model > & s, rng_t & rng )`

Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise1DObjective](#).

#### Parameters

in	<i>model</i>	Model object
in	<i>s</i>	The stencil computed at theta.
in, out	<i>rng</i>	A random number generator

#### Returns

A simulated image at theta under the noise model.

Definition at line 45 of file `PoissonNoise1DObjective.h`.

References `mappel::generate_poisson()`.

8.5.2.38 `template<class Model , class rng_t > ReturnIfSubclassT<ImageT<Model>, Model, PoissonNoise2DObjective> mappel::methods::simulate_image ( const Model & model, const StencilT< Model > & s, rng_t & rng )`

Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise2DObjective](#).

**Parameters**

in	<i>model</i>	Model object
in	<i>s</i>	The stencil computed at theta.
in, out	<i>rng</i>	A random number generator

**Returns**

A simulated image at theta under the noise model.

Definition at line 45 of file PoissonNoise2DObjective.h.

References mappel::generate\_poisson(), and mappel::ImageFormat2DBase::size.

8.5.2.39 `template<class Model , class rng_t > ModelDataT<Model> mappel::methods::simulate_image ( const Model & model, const ParamT< Model > & theta, rng_t & rng )`

8.5.2.40 `template<class Model > ModelDataT< Model > mappel::methods::simulate_image ( const Model & model, const StencilT< Model > & s )`

Definition at line 35 of file model\_methods\_impl.h.

References simulate\_image().

8.5.2.41 `template<class Model > ModelDataT< Model > mappel::methods::simulate_image_from_model ( const Model & model, const ImageT< Model > & model_im )`

Definition at line 41 of file model\_methods\_impl.h.

8.5.2.42 `template<class Model , class rng_t > ReturnIfSubclassT<ModelDataT<Model>, Model, PoissonNoise1DObjective> mappel::methods::simulate_image_from_model ( const Model & model, const ImageT< Model > & model_im, rng_t & rng )`

Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise1DObjective](#).

**Parameters**

in	<i>model</i>	Model object
in	<i>model_im</i>	An image representing the expected (mean) at each pixel under the PSF model.
in, out	<i>rng</i>	A random number generator

**Returns**

A simulated image corresponding to model\_im under the noise model.

Definition at line 61 of file PoissonNoise1DObjective.h.

References mappel::generate\_poisson().

8.5.2.43 `template<class Model , class rng_t > ReturnIfSubclassT<ImageT<Model>, Model, PoissonNoise2DObjective> mappel::methods::simulate_image_from_model ( const Model & model, const ImageT< Model > & model_im, rng_t & rng )`

Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise2DObjective](#).

#### Parameters

in	<i>model</i>	Model object
in	<i>model_im</i>	An image representing the expected (mean) at each pixel under the PSF model.
in, out	<i>rng</i>	A random number generator

#### Returns

A simulated image corresponding to *model\_im* under the noise model.

Definition at line 64 of file `PoissonNoise2DObjective.h`.

References `mappel::generate_poisson()`, and `mappel::ImageFormat2DBase::size`.

## 8.6 mappel::methods::debug Namespace Reference

### Functions

- `template<class Model > void estimate_max_debug (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, estimator::MLEDebugData &mle, StatsT &stats)`
- `template<class Model > void error_bounds_profile_likelihood_debug (const Model &model, const ModelDataT< Model > &data, estimator::ProfileBoundsDebugData &bounds, StatsT &stats)`
- `template<class Model > void estimate_posterior_debug (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, mcmc::MCMCDebugData &mcmc_debug_sample)`

#### 8.6.1 Function Documentation

8.6.1.1 `template<class Model > void mappel::methods::debug::error_bounds_profile_likelihood_debug ( const Model & model, const ModelDataT< Model > & data, estimator::ProfileBoundsDebugData & bounds, StatsT & stats )`

Definition at line 461 of file `model_methods_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, and `mappel::estimator::IterativeMaximizer< Model >::get_stats()`.

8.6.1.2 `template<class Model > void mappel::methods::debug::estimate_max_debug ( const Model & model, const ModelDataT< Model > & data, const std::string & method, const ParamT< Model > & theta_init, estimator::MLEDebugData & mle, StatsT & stats )`

Definition at line 452 of file `model_methods_impl.h`.

References `mappel::methods::make_estimator()`.

Referenced by `mappel::estimator::Estimator< Model >::~~Estimator()`.

8.6.1.3 `template<class Model > void mappel::methods::debug::estimate_posterior_debug ( const Model & model, const ModelDataT< Model > & data, const ParamT< Model > & theta_init, mcmc::MCMCDebugData & mcmc_debug_sample )`

Definition at line 470 of file `model_methods_impl.h`.

References `mappel::mcmc::MCMCDebugData::candidate`, `mappel::mcmc::MCMCDebugData::candidate_rllh`, `mappel::mcmc::MCMCDebugData::initialize_arrays()`, `mappel::mcmc::MCMCDebugData::Nsample`, `mappel::mcmc::MCMCDebugData::sample`, `mappel::mcmc::sample_posterior_debug()`, and `mappel::mcmc::MCMCDebugData::sample_rllh`.

## 8.7 mappel::methods::likelihood Namespace Reference

### Namespaces

- [debug](#)

### Functions

- `template<class Model > ReturnIfSubclassT< double, Model, PoissonNoise1DObjective > llh (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model > ReturnIfSubclassT< double, Model, PoissonNoise1DObjective > rllh (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model > ReturnIfSubclassT< ParamT< Model >, Model, PoissonNoise1DObjective > grad (const Model &model, const ModelDataT< Model > &im, const StencilT< Model > &s)`
- `template<class Model > ReturnIfSubclassT< void, Model, PoissonNoise1DObjective > grad2 (const Model &model, const ModelDataT< Model > &im, const StencilT< Model > &s, ParamT< Model > &grad_val, ParamT< Model > &grad2_val)`
- `template<class Model > ReturnIfSubclassT< void, Model, PoissonNoise1DObjective > hessian (const Model &model, const ModelDataT< Model > &im, const StencilT< Model > &s, ParamT< Model > &grad_val, MatT &hess_val)`
- `template<class Model > ReturnIfSubclassT< double, Model, PoissonNoise2DObjective > llh (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model > ReturnIfSubclassT< double, Model, PoissonNoise2DObjective > rllh (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`

- `template<class Model >`  
`ReturnIfSubclassT< ParamT< Model >, Model, PoissonNoise2DObjective > grad (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, PoissonNoise2DObjective > grad2 (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad_val, ParamT< Model > &grad2_val)`
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, PoissonNoise2DObjective > hessian (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad_val, MatT &hess_val)`

### 8.7.1 Function Documentation

**8.7.1.1** `template<class Model > ReturnIfSubclassT<ParamT<Model>,Model,PoissonNoise1DObjective>`  
`mappel::methods::likelihood::grad ( const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & s )`

Definition at line 146 of file PoissonNoise1DObjective.h.

**8.7.1.2** `template<class Model > ReturnIfSubclassT<ParamT<Model>,Model,PoissonNoise2DObjective>`  
`mappel::methods::likelihood::grad ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 159 of file PoissonNoise2DObjective.h.

References `mappel::ImageFormat2DBase::size`.

**8.7.1.3** `template<class Model > ReturnIfSubclassT<void,Model,PoissonNoise1DObjective>`  
`mappel::methods::likelihood::grad2 ( const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & s, ParamT< Model > & grad_val, ParamT< Model > & grad2_val )`

Definition at line 163 of file PoissonNoise1DObjective.h.

**8.7.1.4** `template<class Model > ReturnIfSubclassT<void,Model,PoissonNoise2DObjective>`  
`mappel::methods::likelihood::grad2 ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s, ParamT< Model > & grad_val, ParamT< Model > & grad2_val )`

Definition at line 177 of file PoissonNoise2DObjective.h.

References `mappel::ImageFormat2DBase::size`.

**8.7.1.5** `template<class Model > ReturnIfSubclassT<void,Model,PoissonNoise1DObjective>`  
`mappel::methods::likelihood::hessian ( const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & s, ParamT< Model > & grad_val, MatT & hess_val )`

Definition at line 186 of file PoissonNoise1DObjective.h.

Referenced by `mappel::methods::aposteriori_objective()`, and `mappel::methods::likelihood_objective()`.

**8.7.1.6** `template<class Model > ReturnIfSubclassT<void,Model,PoissonNoise2DObjective>  
 mappel::methods::likelihood::hessian ( const Model & model, const ModelDataT< Model > & data_im, const  
 StencilT< Model > & s, ParamT< Model > & grad_val, MatT & hess_val )`

Definition at line 202 of file PoissonNoise2DObjective.h.

References mappel::ImageFormat2DBase::size.

**8.7.1.7** `template<class Model > ReturnIfSubclassT<double,Model,PoissonNoise1DObjective>  
 mappel::methods::likelihood::llh ( const Model & model, const ModelDataT< Model > & data_im, const StencilT<  
 Model > & s )`

Definition at line 122 of file PoissonNoise1DObjective.h.

References mappel::poisson\_log\_likelihood().

**8.7.1.8** `template<class Model > ReturnIfSubclassT<double,Model,PoissonNoise2DObjective>  
 mappel::methods::likelihood::llh ( const Model & model, const ModelDataT< Model > & data_im, const StencilT<  
 Model > & s )`

Definition at line 131 of file PoissonNoise2DObjective.h.

References mappel::poisson\_log\_likelihood(), and mappel::ImageFormat2DBase::size.

**8.7.1.9** `template<class Model > ReturnIfSubclassT<double,Model,PoissonNoise1DObjective>  
 mappel::methods::likelihood::rllh ( const Model & model, const ModelDataT< Model > & data_im, const StencilT<  
 Model > & s )`

Definition at line 134 of file PoissonNoise1DObjective.h.

References mappel::relative\_poisson\_log\_likelihood().

Referenced by mappel::methods::aposteriori\_objective(), and mappel::methods::likelihood\_objective().

**8.7.1.10** `template<class Model > ReturnIfSubclassT<double,Model,PoissonNoise2DObjective>  
 mappel::methods::likelihood::rllh ( const Model & model, const ModelDataT< Model > & data_im, const StencilT<  
 Model > & s )`

Definition at line 145 of file PoissonNoise2DObjective.h.

References mappel::relative\_poisson\_log\_likelihood(), and mappel::ImageFormat2DBase::size.



## 8.8 mappel::methods::likelihood::debug Namespace Reference

### Functions

- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, PoissonNoise1DObjective > llh_components` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, PoissonNoise1DObjective > rllh_components` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, PoissonNoise1DObjective > grad_components` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- `template<class Model >`  
`ReturnIfSubclassT< CubeT, Model, PoissonNoise1DObjective > hessian_components` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, PoissonNoise2DObjective > llh_components` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, PoissonNoise2DObjective > rllh_components` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, PoissonNoise2DObjective > grad_components` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- `template<class Model >`  
`ReturnIfSubclassT< CubeT, Model, PoissonNoise2DObjective > hessian_components` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)

### 8.8.1 Function Documentation

**8.8.1.1** `template<class Model > ReturnIfSubclassT<MatT,Model,PoissonNoise1DObjective>`  
`mappel::methods::likelihood::debug::grad_components` ( const Model & *model*, const ModelDataT< Model > & *data\_im*, const StencilT< Model > & *s* )

Definition at line 230 of file PoissonNoise1DObjective.h.

**8.8.1.2** `template<class Model > ReturnIfSubclassT<MatT,Model,PoissonNoise2DObjective>`  
`mappel::methods::likelihood::debug::grad_components` ( const Model & *model*, const ModelDataT< Model > & *data\_im*, const StencilT< Model > & *s* )

Definition at line 255 of file PoissonNoise2DObjective.h.

References `mappel::ImageFormat2DBase::size`.

**8.8.1.3** `template<class Model > ReturnIfSubclassT<CubeT,Model,PoissonNoise1DObjective>`  
`mappel::methods::likelihood::debug::hessian_components` ( const Model & *model*, const ModelDataT< Model > & *data\_im*, const StencilT< Model > & *s* )

Definition at line 246 of file PoissonNoise1DObjective.h.

```
8.8.1.4  template<class Model > ReturnIfSubclassT<CubeT,Model,PoissonNoise2DObjective>
        mappel::methods::likelihood::debug::hessian_components ( const Model & model, const ModelDataT< Model > &
        data_im, const StencilT< Model > & s )
```

Definition at line 274 of file PoissonNoise2DObjective.h.

References mappel::ImageFormat2DBase::size.

```
8.8.1.5  template<class Model > ReturnIfSubclassT<VecT,Model,PoissonNoise1DObjective>
        mappel::methods::likelihood::debug::llh_components ( const Model & model, const ModelDataT< Model > & data_im,
        const StencilT< Model > & s )
```

Definition at line 206 of file PoissonNoise1DObjective.h.

References mappel::poisson\_log\_likelihood().

```
8.8.1.6  template<class Model > ReturnIfSubclassT<VecT,Model,PoissonNoise2DObjective>
        mappel::methods::likelihood::debug::llh_components ( const Model & model, const ModelDataT< Model > & data_im,
        const StencilT< Model > & s )
```

Definition at line 225 of file PoissonNoise2DObjective.h.

References mappel::poisson\_log\_likelihood(), and mappel::ImageFormat2DBase::size.

```
8.8.1.7  template<class Model > ReturnIfSubclassT<VecT,Model,PoissonNoise1DObjective>
        mappel::methods::likelihood::debug::rllh_components ( const Model & model, const ModelDataT< Model > & data_im,
        const StencilT< Model > & s )
```

Definition at line 218 of file PoissonNoise1DObjective.h.

References mappel::relative\_poisson\_log\_likelihood().

```
8.8.1.8  template<class Model > ReturnIfSubclassT<VecT,Model,PoissonNoise2DObjective>
        mappel::methods::likelihood::debug::rllh_components ( const Model & model, const ModelDataT< Model > & data_im,
        const StencilT< Model > & s )
```

Definition at line 240 of file PoissonNoise2DObjective.h.

References mappel::relative\_poisson\_log\_likelihood(), and mappel::ImageFormat2DBase::size.

## 8.9 mappel::methods::objective Namespace Reference

### Namespaces

- [debug](#)
- [openmp](#)

## Functions

- `template<class Model >`  
`ReturnIfSubclassT< double, Model, MAPEstimator > llh` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< double, Model, MAPEstimator > rllh` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< ParamT< Model >, Model, MAPEstimator > grad` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, MAPEstimator > grad2` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s, `ParamT< Model >` &grad, `ParamT< Model >` &grad2)
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, MAPEstimator > hessian` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s, `ParamT< Model >` &grad, `MatT` &hess)
- `template<class Model >`  
`ReturnIfSubclassT< double, Model, MLEstimator > llh` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< double, Model, MLEstimator > rllh` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< ParamT< Model >, Model, MLEstimator > grad` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, MLEstimator > grad2` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s, `ParamT< Model >` &grad, `ParamT< Model >` &grad2)
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, MLEstimator > hessian` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s, `ParamT< Model >` &grad, `MatT` &hess)
- `template<class Model >`  
`double llh` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)
- `template<class Model >`  
`double rllh` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)
- `template<class Model >`  
`ParamT< Model > grad` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)
- `template<class Model >`  
`ParamT< Model > grad2` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)
- `template<class Model >`  
`void grad2` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta, `ParamT< Model >` &grad\_val, `ParamT< Model >` &grad2\_val)
- `template<class Model >`  
`MatT hessian` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)
- `template<class Model >`  
`MatT hessian` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`void hessian` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta, `ParamT< Model >` &grad, `MatT` &hess)

- `template<class Model >`  
`void hessian (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, MatT &hess)`
- `template<class Model >`  
`MatT negative\_definite\_hessian (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`MatT negative\_definite\_hessian (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`void negative\_definite\_hessian (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, ParamT< Model > &grad, MatT &hess)`
- `template<class Model >`  
`void negative\_definite\_hessian (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad, MatT &hess)`

### 8.9.1 Function Documentation

**8.9.1.1** `template<class Model > ReturnIfSubclassT<ParamT<Model>, Model,MLEstimator>`  
`mappel::methods::objective::grad ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 51 of file MLEstimator.h.

**8.9.1.2** `template<class Model > ReturnIfSubclassT<ParamT<Model>,Model,MAPEstimator>`  
`mappel::methods::objective::grad ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 55 of file MAPEstimator.h.

Referenced by `mappel::methods::aposteriori_objective()`, `grad()`, `mappel::methods::objective::openmp::grad_stack()`, `hessian()`, `mappel::methods::likelihood_objective()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile←_maximize()`, `negative_definite_hessian()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound←_backtrack()`.

**8.9.1.3** `template<class Model > ParamT< Model > mappel::methods::objective::grad ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta )`

Definition at line 65 of file model\_methods\_impl.h.

References `grad()`.

**8.9.1.4** `template<class Model > ReturnIfSubclassT<void,Model,MLEstimator> mappel::methods::objective::grad2 ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s, ParamT< Model > & grad, ParamT< Model > & grad2 )`

Definition at line 58 of file MLEstimator.h.

8.9.1.5 `template<class Model > ReturnIfSubclassT<void,Model,MAPEstimator> mappel::methods::objective::grad2 ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s, ParamT< Model > & grad, ParamT< Model > & grad2 )`

Definition at line 64 of file MAPEstimator.h.

Referenced by `grad2()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

8.9.1.6 `template<class Model > ParamT< Model > mappel::methods::objective::grad2 ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta )`

Definition at line 77 of file model\_methods\_impl.h.

References `grad2()`.

8.9.1.7 `template<class Model > void mappel::methods::objective::grad2 ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta, ParamT< Model > & grad_val, ParamT< Model > & grad2_val )`

Definition at line 92 of file model\_methods\_impl.h.

References `grad2()`.

8.9.1.8 `template<class Model > ReturnIfSubclassT<void,Model,MLEstimator> mappel::methods::objective::hessian ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s, ParamT< Model > & grad, MatT & hess )`

Definition at line 65 of file MLEstimator.h.

8.9.1.9 `template<class Model > MatT mappel::methods::objective::hessian ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta )`

Definition at line 100 of file model\_methods\_impl.h.

References `hessian()`.

8.9.1.10 `template<class Model > ReturnIfSubclassT<void,Model,MAPEstimator> mappel::methods::objective::hessian ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s, ParamT< Model > & grad, MatT & hess )`

Definition at line 72 of file MAPEstimator.h.

Referenced by `hessian()`, `mappel::methods::objective::openmp::hessian_stack()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `negative_definite_hessian()`, and `mappel::methods::observed_information()`.

8.9.1.11 `template<class Model > MatT mappel::methods::objective::hessian ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 112 of file model\_methods\_impl.h.

References `grad()`, and `hessian()`.

8.9.1.12 `template<class Model > void mappel::methods::objective::hessian ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta, ParamT< Model > & grad, MatT & hess )`

Definition at line 122 of file `model_methods_impl.h`.

References `grad()`, and `hessian()`.

8.9.1.13 `template<class Model > void mappel::methods::objective::hessian ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta, MatT & hess )`

Definition at line 134 of file `model_methods_impl.h`.

References `grad()`, and `hessian()`.

8.9.1.14 `template<class Model > ReturnIfSubclassT<double,Model,MAPEstimator> mappel::methods::objective::llh ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 36 of file `MAPEstimator.h`.

Referenced by `llh()`, `mappel::methods::objective::openmp::llh_stack()`, and `log_likelihood()`.

8.9.1.15 `template<class Model > ReturnIfSubclassT<double,Model,MLEstimator> mappel::methods::objective::llh ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 37 of file `MLEstimator.h`.

8.9.1.16 `template<class Model > double mappel::methods::objective::llh ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta )`

Definition at line 49 of file `model_methods_impl.h`.

References `llh()`.

8.9.1.17 `template<class Model > MatT mappel::methods::objective::negative_definite_hessian ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta )`

Definition at line 142 of file `model_methods_impl.h`.

Referenced by `negative_definite_hessian()`, and `mappel::methods::objective::openmp::negative_definite_hessian_stack()`.

8.9.1.18 `template<class Model > MatT mappel::methods::objective::negative_definite_hessian ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 154 of file `model_methods_impl.h`.

References `grad()`, and `negative_definite_hessian()`.

8.9.1.19 `template<class Model > void mappel::methods::objective::negative_definite_hessian ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta, ParamT< Model > & grad, MatT & hess )`

Definition at line 164 of file `model_methods_impl.h`.

References `grad()`, and `negative_definite_hessian()`.

8.9.1.20 `template<class Model > void mappel::methods::objective::negative_definite_hessian ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s, ParamT< Model > & grad, MatT & hess )`

Definition at line 177 of file `model_methods_impl.h`.

References `mappel::cholesky_make_negative_definite()`, and `hessian()`.

8.9.1.21 `template<class Model > ReturnIfSubclassT<double,Model,MLEstimator> mappel::methods::objective::rllh ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 44 of file `MLEstimator.h`.

8.9.1.22 `template<class Model > ReturnIfSubclassT<double,Model,MAPEstimator> mappel::methods::objective::rllh ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 48 of file `MAPEstimator.h`.

Referenced by `mappel::methods::aposteriori_objective()`, `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::methods::likelihood_objective()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::MaximizerData()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `relative_log_likelihood()`, `rllh()`, `mappel::methods::objective::openmp::rllh_stack()`, `mappel::mcmc::sample_posterior()`, and `mappel::mcmc::sample_posterior_debug()`.

8.9.1.23 `template<class Model > double mappel::methods::objective::rllh ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta )`

Definition at line 57 of file `model_methods_impl.h`.

References `rllh()`.

## 8.10 mappel::methods::objective::debug Namespace Reference

### Functions

- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, MAPEstimator > llh_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, MAPEstimator > rllh_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, MAPEstimator > grad_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< CubeT, Model, MAPEstimator > hessian_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, MLEstimator > llh_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, MLEstimator > rllh_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, MLEstimator > grad_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`ReturnIfSubclassT< CubeT, Model, MLEstimator > hessian_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `StencilT< Model >` &s)
- `template<class Model >`  
`VecT llh_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)
- `template<class Model >`  
`VecT rllh_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)
- `template<class Model >`  
`MatT grad_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)
- `template<class Model >`  
`CubeT hessian_components` (const Model &model, const `ModelDataT< Model >` &data\_im, const `ParamT< Model >` &theta)

### 8.10.1 Function Documentation

- 8.10.1.1 `template<class Model > ReturnIfSubclassT<MatT,Model,MLEstimator> mappel::methods::objective::debug::grad_components ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 88 of file MLEstimator.h.



8.10.1.2 `template<class Model> ReturnIfSubclassT<MatT,Model,MAPEstimator> mappel::methods::objective::debug::grad_components ( const Model & model, const ModelDataT< Model> & data_im, const StencilT< Model> & s )`

Definition at line 96 of file MAPEstimator.h.

Referenced by `grad_components()`.

8.10.1.3 `template<class Model> MatT mappel::methods::objective::debug::grad_components ( const Model & model, const ModelDataT< Model> & data_im, const ParamT< Model> & theta )`

Definition at line 201 of file model\_methods\_impl.h.

References `grad_components()`.

8.10.1.4 `template<class Model> ReturnIfSubclassT<CubeT,Model,MLEstimator> mappel::methods::objective::debug::hessian_components ( const Model & model, const ModelDataT< Model> & data_im, const StencilT< Model> & s )`

Definition at line 95 of file MLEstimator.h.

8.10.1.5 `template<class Model> ReturnIfSubclassT<CubeT,Model,MAPEstimator> mappel::methods::objective::debug::hessian_components ( const Model & model, const ModelDataT< Model> & data_im, const StencilT< Model> & s )`

Definition at line 103 of file MAPEstimator.h.

Referenced by `hessian_components()`.

8.10.1.6 `template<class Model> CubeT mappel::methods::objective::debug::hessian_components ( const Model & model, const ModelDataT< Model> & data_im, const ParamT< Model> & theta )`

Definition at line 208 of file model\_methods\_impl.h.

References `hessian_components()`.

8.10.1.7 `template<class Model> ReturnIfSubclassT<VecT,Model,MLEstimator> mappel::methods::objective::debug::llh_components ( const Model & model, const ModelDataT< Model> & data_im, const StencilT< Model> & s )`

Definition at line 74 of file MLEstimator.h.

8.10.1.8 `template<class Model> ReturnIfSubclassT<VecT,Model,MAPEstimator> mappel::methods::objective::debug::llh_components ( const Model & model, const ModelDataT< Model> & data_im, const StencilT< Model> & s )`

Definition at line 82 of file MAPEstimator.h.

Referenced by `llh_components()`.

8.10.1.9 `template<class Model > VecT mappel::methods::objective::debug::llh_components ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta )`

Definition at line 187 of file `model_methods_impl.h`.

References `llh_components()`.

8.10.1.10 `template<class Model > ReturnIfSubclassT<VecT,Model,MLEstimator> mappel::methods::objective::debug::rllh_components ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 81 of file `MLEstimator.h`.

8.10.1.11 `template<class Model > ReturnIfSubclassT<VecT,Model,MAPEstimator> mappel::methods::objective::debug::rllh_components ( const Model & model, const ModelDataT< Model > & data_im, const StencilT< Model > & s )`

Definition at line 89 of file `MAPEstimator.h`.

Referenced by `rllh_components()`.

8.10.1.12 `template<class Model > VecT mappel::methods::objective::debug::rllh_components ( const Model & model, const ModelDataT< Model > & data_im, const ParamT< Model > & theta )`

Definition at line 194 of file `model_methods_impl.h`.

References `rllh_components()`.

## 8.11 mappel::methods::objective::openmp Namespace Reference

### Functions

- `template<class Model > void llh_stack (const Model &model, const ImageT< Model > &image, const ParamVecT< Model > &theta_stack, VecT &llh_stack)`  
*Parallel log\_likelihood calculations for a single image.*
- `template<class Model > void llh_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, VecT &llh_stack)`  
*Parallel log\_likelihood calculations for a stack of images.*
- `template<class Model > void rllh_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, VecT &rllh_stack)`  
*Parallel relative log\_likelihood calculations for a stack of images.*
- `template<class Model > void rllh_stack (const Model &model, const ImageT< Model > &image, const ParamVecT< Model > &theta_stack, VecT &rllh_stack)`

- `template<class Model >`  
`void grad\_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, ParamVecT< Model > &grad_stack)`  
*Parallel model gradient calculations for a stack of images.*
- `template<class Model >`  
`void hessian\_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, CubeT &hessian_stack)`  
*Parallel model Hessian calculations for a stack of images.*
- `template<class Model >`  
`void negative\_definite\_hessian\_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, CubeT &hessian_stack)`  
*Parallel model negative\_definite Hessian approximation calculations for a stack of images.*

### 8.11.1 Function Documentation

**8.11.1.1** `template<class Model > void mappel::methods::objective::openmp::grad_stack ( const Model & model, const ImageStackT< Model > & image_stack, const ParamVecT< Model > & theta_stack, ParamVecT< Model > & grad_stack )`

Parallel model gradient calculations for a stack of images.

Compute gradient of log-likelihood for multiple image, theta pairs.

Use: `model.make_param_stack()` to make a parameter stack of appropriate dimensions for the model gradients.

#### Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

#### Parameters

in	<i>model</i>	A <a href="#">PointEmitterModel</a> object.
in	<i>image_stack</i>	Sequence of images.
in	<i>theta_stack</i>	Sequence of thetas.
out	<i>grad_stack</i>	Sequence of grad vectors values computed. Size: [model.num_params, n]

Definition at line 504 of file `openmp_methods.h`.

References `mappel::methods::objective::grad()`, `omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::run()`.

**8.11.1.2** `template<class Model > void mappel::methods::objective::openmp::hessian_stack ( const Model & model, const ImageStackT< Model > & image_stack, const ParamVecT< Model > & theta_stack, CubeT & hessian_stack )`

Parallel model Hessian calculations for a stack of images.

Compute Hessian of log-likelihood for multiple image, theta pairs.

Use: `model.make_param_mat_stack()` to make a parameter matrix stack of appropriate dimensions for the model Hessian.

### Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

### Parameters

in	<i>model</i>	A <a href="#">PointEmitterModel</a> object.
in	<i>image_stack</i>	Sequence of images.
in	<i>theta_stack</i>	Sequence of thetas. Size: [model.num_params, nThetas]
out	<i>hess_stack</i>	Sequence of Hessian matrices computed. Size: [model.num_params, model.num_params, n]

Definition at line 552 of file `openmp_methods.h`.

References `mappel::methods::objective::hessian()`, `omp_exception_catcher::impl::OMPExcceptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl::OMPExcceptionCatcher<_dummy>::run()`.

**8.11.1.3** `template<class Model > void mappel::methods::objective::openmp::llh_stack ( const Model & model, const ImageT<Model > & image, const ParamVecT<Model > & theta_stack, VecT & llh_stack )`

Parallel `log_likelihood` calculations for a single image.

Compute log-likelihood for multiple thetas using the same image

Use: `model.make_param_stack()` to make a parameter stack of appropriate dimensions for the model

### Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

### Parameters

in	<i>model</i>	A <a href="#">PointEmitterModel</a> object.
in	<i>image</i>	An image.
in	<i>theta_stack</i>	Sequence of thetas. Size: [model.num_params, nThetas]
out	<i>llh_stack</i>	Sequence of llh values computed.

Definition at line 368 of file `openmp_methods.h`.

References `omp_exception_catcher::impl::OMPExcceptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl::OMPExcceptionCatcher<_dummy>::run()`.

**8.11.1.4** `template<class Model > void mappel::methods::objective::openmp::llh_stack ( const Model & model, const ImageStackT<Model > & image_stack, const ParamVecT<Model > & theta_stack, VecT & llh_stack )`

Parallel `log_likelihood` calculations for a stack of images.

Compute log-likelihood for multiple image, theta pairs.

Use: `model.make_param_stack()` to make a parameter stack of appropriate dimensions for the model

## Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

## Parameters

in	<i>model</i>	A <a href="#">PointEmitterModel</a> object.
in	<i>image_stack</i>	Sequence of images.
in	<i>theta_stack</i>	Sequence of thetas. Size: [model.num_params, nThetas]
out	<i>llh_stack</i>	Sequence of llh values computed. Size: [n]

Definition at line 393 of file openmp\_methods.h.

References [mappel::methods::objective::llh\(\)](#), [omp\\_exception\\_catcher::impl::OMPExcceptionCatcher<\\_dummy>::rethrow\(\)](#), and [omp\\_exception\\_catcher::impl::OMPExcceptionCatcher<\\_dummy>::run\(\)](#).

**8.11.1.5** `template<class Model> void mappel::methods::objective::openmp::negative_definite_hessian_stack ( const Model & model, const ImageStackT< Model > & image_stack, const ParamVecT< Model > & theta_stack, CubeT & hessian_stack )`

Parallel model negative\_definite Hessian approximation calculations for a stack of images.

Compute Hessian a negative\_definite Hessian using a modified Cholesky decompositions. Computes for multiple image, theta pairs.

Use: [model.make\\_param\\_mat\\_stack\(\)](#) to make a parameter matrix stack of appropriate dimensions for the model Hessian.

## Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

## Parameters

in	<i>model</i>	A <a href="#">PointEmitterModel</a> object.
in	<i>image_stack</i>	Sequence of images.
in	<i>theta_stack</i>	Sequence of thetas. Size: [model.num_params, nThetas]
out	<i>hess_stack</i>	Sequence of approximate Hessian negative definite matrices computed. Size: [model.num_params, model.num_params, n]

Definition at line 601 of file openmp\_methods.h.

References [mappel::methods::objective::negative\\_definite\\_hessian\(\)](#), [omp\\_exception\\_catcher::impl::OMPExcceptionCatcher<\\_dummy>::rethrow\(\)](#), and [omp\\_exception\\_catcher::impl::OMPExcceptionCatcher<\\_dummy>::run\(\)](#).

**8.11.1.6** `template<class Model> void mappel::methods::objective::openmp::rllh_stack ( const Model & model, const ImageStackT< Model > & image_stack, const ParamVecT< Model > & theta_stack, VecT & rllh_stack )`

Parallel relative log\_likelihood calculations for a stack of images.

Compute relative log-likelihood for multiple image, theta pairs.

Use: `model.make_param_stack()` to make a parameter stack of appropriate dimensions for the model

#### Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

#### Parameters

in	<i>model</i>	A <a href="#">PointEmitterModel</a> object.
in	<i>image_stack</i>	Sequence of images.
in	<i>theta_stack</i>	Sequence of thetas. Size: [model.num_params, nThetas]
out	<i>rllh_stack</i>	Sequence of rllh values computed. Size: [n]

Definition at line 440 of file `openmp_methods.h`.

References `omp_exception_catcher::impl_::OMPEXceptionCatcher<_dummy>::rethrow()`, `mappel::methods::objective::rllh()`, `rllh_stack()`, and `omp_exception_catcher::impl_::OMPEXceptionCatcher<_dummy>::run()`.

Referenced by `mappel::estimator::CGaussMLE<Model>::get_debug_stats()`.

**8.11.1.7** `template<class Model> void mappel::methods::objective::openmp::rllh_stack ( const Model & model, const ImageT<Model> & image, const ParamVecT<Model> & theta_stack, VecT & rllh_stack )`

Definition at line 476 of file `openmp_methods.h`.

References `omp_exception_catcher::impl_::OMPEXceptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl_::OMPEXceptionCatcher<_dummy>::run()`.

Referenced by `rllh_stack()`.

## 8.12 mappel::methods::openmp Namespace Reference

#### Functions

- `template<class Model>`  
void [sample\\_prior\\_stack](#) (const Model &model, [ParamVecT](#)<Model> &theta\_stack)  
*Parallel sampling of the model prior.*
- `template<class Model>`  
void [model\\_image\\_stack](#) (const Model &model, const [ParamVecT](#)<Model> &theta\_stack, [ImageStackT](#)<Model> &image\_stack)  
*Parallel computation of the model image.*
- `template<class Model>`  
void [simulate\\_image\\_stack](#) (const Model &model, const [ParamVecT](#)<Model> &theta\_stack, [ImageStackT](#)<Model> &image\_stack)  
*Parallel simulation of images from one or more theta.*

- `template<class Model >`  
`void cr_lower_bound_stack (const Model &model, const ParamVecT< Model > &theta_stack, ParamVecT< Model > &crlb_stack)`
- `template<class Model >`  
`void expected_information_stack (const Model &model, const ParamVecT< Model > &theta_stack, CubeT &fisherl_stack)`
- `template<class Model >`  
`void estimate_max_stack (const Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, estimator::MLEDataStack &mle_data_stack)`
- `template<class Model >`  
`void estimate_max_stack (const Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_init_stack, estimator::MLEDataStack &mle_data_stack)`
- `template<class Model >`  
`void estimate_max_stack (const Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_init_stack, estimator::MLEDataStack &mle_data_stack, StatsT &stats)`
- `template<class Model >`  
`void estimate_profile_likelihood_stack (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamVecT< Model > &fixed_theta_init, estimator::ProfileLikelihoodData &est)`
- `template<class Model >`  
`void estimate_profile_likelihood_stack (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamVecT< Model > &fixed_theta_init, estimator::ProfileLikelihoodData &est, StatsT &stats)`
- `template<class Model >`  
`void estimate_posterior_stack (const Model &model, const ModelDataStackT< Model > &data_stack, const ParamVecT< Model > &theta_init_stack, mcmc::MCMCDataStack &est)`
- `template<class Model >`  
`void estimate_posterior_stack (const Model &model, const ModelDataStackT< Model > &data_stack, mcmc::MCMCDataStack &est)`
- `template<class Model >`  
`void error_bounds_expected_stack (const Model &model, const MatT &theta_est_stack, double confidence, MatT &theta_lb_stack, MatT &theta_ub_stack)`
- `template<class Model >`  
`void error_bounds_observed_stack (const Model &model, const MatT &theta_est_stack, CubeT &obsl_stack, double confidence, MatT &theta_lb_stack, MatT &theta_ub_stack)`
- `template<class Model >`  
`void error_bounds_profile_likelihood_parallel (const Model &model, const ModelDataStackT< Model > &image, estimator::ProfileBoundsData &est, StatsT &stats)`
- `template<class Model >`  
`void error_bounds_profile_likelihood_parallel (const Model &model, const ModelDataT< Model > &image, estimator::ProfileBoundsData &est)`
- `template<class Model >`  
`void error_bounds_profile_likelihood_stack (const Model &model, const ModelDataStackT< Model > &image, estimator::ProfileBoundsDataStack &est, StatsT &stats)`
- `template<class Model >`  
`void error_bounds_profile_likelihood_stack (const Model &model, const ModelDataStackT< Model > &image, estimator::ProfileBoundsDataStack &est)`

### 8.12.1 Function Documentation

**8.12.1.1** `template<class Model> void mappel::methods::openmp::cr_lower_bound_stack ( const Model & model, const ParamVecT< Model> & theta_stack, ParamVecT< Model> & crlb_stack )`

Definition at line 138 of file openmp\_methods.h.

References `omp_exception_catcher::impl::OMPEXceptionCatcher< _dummy>::rethrow()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< _dummy>::run()`.

**8.12.1.2** `template<class Model> void mappel::methods::openmp::error_bounds_expected_stack ( const Model & model, const MatT & theta_est_stack, double confidence, MatT & theta_lb_stack, MatT & theta_ub_stack )`

Definition at line 262 of file openmp\_methods.h.

References `mappel::normal_quantile_twosided()`, `omp_exception_catcher::impl::OMPEXceptionCatcher< _dummy>::rethrow()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< _dummy>::run()`.

**8.12.1.3** `template<class Model> void mappel::methods::openmp::error_bounds_observed_stack ( const Model & model, const MatT & theta_est_stack, CubeT & obsl_stack, double confidence, MatT & theta_lb_stack, MatT & theta_ub_stack )`

Definition at line 283 of file openmp\_methods.h.

References `mappel::normal_quantile_twosided()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< _dummy>::run()`.

**8.12.1.4** `template<class Model> void mappel::methods::openmp::error_bounds_profile_likelihood_parallel ( const Model & model, const ModelDataStackT< Model> & image, estimator::ProfileBoundsData & est, StatsT & stats )`

Profile likelihood bounds. Uses the Venzon and Moolgavkar (VM) algorithm for computing each of the bounds of the profile likelihood.

Definition at line 312 of file openmp\_methods.h.

References `mappel::chisq_quantile()`, `mappel::estimator::ProfileBoundsData::confidence`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_parallel()`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::IterativeMaximizer< Model>::get_stats()`, and `mappel::estimator::ProfileBoundsData::target_rllh_delta`.

**8.12.1.5** `template<class Model> void mappel::methods::openmp::error_bounds_profile_likelihood_parallel ( const Model & model, const ModelDataT< Model> & image, estimator::ProfileBoundsData & est )`

Definition at line 322 of file openmp\_methods.h.

References `mappel::chisq_quantile()`, `mappel::estimator::ProfileBoundsData::confidence`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_parallel()`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, and `mappel::estimator::ProfileBoundsData::target_rllh_delta`.



8.12.1.6 `template<class Model > void mappel::methods::openmp::error_bounds_profile_likelihood_stack ( const Model & model, const ModelDataStackT< Model > & image, estimator::ProfileBoundsDataStack & est, StatsT & stats )`

Definition at line 331 of file `openmp_methods.h`.

References `mappel::chisq_quantile()`, `mappel::estimator::ProfileBoundsDataStack::confidence`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`s, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::ProfileBoundsDataStack::target_rllh_delta`.

8.12.1.7 `template<class Model > void mappel::methods::openmp::error_bounds_profile_likelihood_stack ( const Model & model, const ModelDataStackT< Model > & image, estimator::ProfileBoundsDataStack & est )`

Definition at line 341 of file `openmp_methods.h`.

References `mappel::chisq_quantile()`, `mappel::estimator::ProfileBoundsDataStack::confidence`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`s, and `mappel::estimator::ProfileBoundsDataStack::target_rllh_delta`.

8.12.1.8 `template<class Model > void mappel::methods::openmp::estimate_max_stack ( const Model & model, const ModelDataStackT< Model > & data_stack, const std::string & method, estimator::MLEDataStack & mle_data_stack )`

Definition at line 168 of file `openmp_methods.h`.

References `mappel::methods::make_estimator()`.

Referenced by `mappel::estimator::Estimator< Model >::~~Estimator()`.

8.12.1.9 `template<class Model > void mappel::methods::openmp::estimate_max_stack ( const Model & model, const ModelDataStackT< Model > & data_stack, const std::string & method, ParamVecT< Model > & theta_init_stack, estimator::MLEDataStack & mle_data_stack )`

Definition at line 178 of file `openmp_methods.h`.

References `mappel::methods::make_estimator()`.

8.12.1.10 `template<class Model > void mappel::methods::openmp::estimate_max_stack ( const Model & model, const ModelDataStackT< Model > & data_stack, const std::string & method, ParamVecT< Model > & theta_init_stack, estimator::MLEDataStack & mle_data_stack, StatsT & stats )`

Definition at line 187 of file `openmp_methods.h`.

References `mappel::methods::make_estimator()`.

8.12.1.11 `template<class Model > void mappel::methods::openmp::estimate_posterior_stack ( const Model & model,  
const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack,  
mcmc::MCMCDataStack & est )`

Definition at line 216 of file `openmp_methods.h`.

References `mappel::mcmc::MCMCDataStack::initialize_arrays()`, `mappel::mcmc::MCMCDataStack::Nburnin`, `mappel::mcmc::MCMCDataStack::Ndata`, `mappel::mcmc::MCMCDataStack::Nsample`, `mappel::mcmc::num_oversample()`, `omp_exception_catcher::impl::OMPEExceptionCatcher< _dummy >::rethrow()`, `omp_exception_catcher::impl::OMPEExceptionCatcher< _dummy >::run()`, and `mappel::mcmc::MCMCDataStack::thin`.

Referenced by `estimate_posterior_stack()`.

8.12.1.12 `template<class Model > void mappel::methods::openmp::estimate_posterior_stack ( const Model & model, const  
ModelDataStackT< Model > & data_stack, mcmc::MCMCDataStack & est )`

Definition at line 254 of file `openmp_methods.h`.

References `estimate_posterior_stack()`.

8.12.1.13 `template<class Model > void mappel::methods::openmp::estimate_profile_likelihood_stack ( const Model & model,  
const ModelDataT< Model > & data, const std::string & method, const ParamVecT< Model > & fixed_theta_init,  
estimator::ProfileLikelihoodData & est )`

Definition at line 197 of file `openmp_methods.h`.

References `mappel::methods::make_estimator()`.

8.12.1.14 `template<class Model > void mappel::methods::openmp::estimate_profile_likelihood_stack ( const Model & model,  
const ModelDataT< Model > & data, const std::string & method, const ParamVecT< Model > & fixed_theta_init,  
estimator::ProfileLikelihoodData & est, StatsT & stats )`

Definition at line 206 of file `openmp_methods.h`.

References `mappel::methods::make_estimator()`.

8.12.1.15 `template<class Model > void mappel::methods::openmp::expected_information_stack ( const Model & model, const  
ParamVecT< Model > & theta_stack, CubeT & fisherl_stack )`

Definition at line 153 of file `openmp_methods.h`.

References `omp_exception_catcher::impl::OMPEExceptionCatcher< _dummy >::rethrow()`, and `omp_exception_catcher::impl::OMPEExceptionCatcher< _dummy >::run()`.

8.12.1.16 `template<class Model > void mappel::methods::openmp::model_image_stack ( const Model & model, const  
ParamVecT< Model > & theta_stack, ImageStackT< Model > & image_stack )`

Parallel computation of the model image.

The model image is the expected photon count at each pixel under parameter `theta`.

Use: `model.make_param_stack()` to make a parameter stack of appropriate dimensions for the model Use: `model.make_image_stack()` to make an image stack of appropriate dimensions for the model

## Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

## Parameters

in	<i>model</i>	A <a href="#">PointEmitterModel</a> object.
in	<i>theta_stack</i>	Sequence of thetas for which to generate images. Size: [model.num_params, nThetas]
out	<i>image_stack</i>	Sequence of model images generated.

Definition at line 73 of file openmp\_methods.h.

References `omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::run()`.

**8.12.1.17** `template<class Model> void mappel::methods::openmp::sample_prior_stack ( const Model & model, ParamVecT<Model> & theta_stack )`

Parallel sampling of the model prior.

Use: `model.make_param_stack()` to make a parameter stack of appropriate dimensions for the model

## Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

## Parameters

in	<i>model,A</i>	<a href="#">PointEmitterModel</a> object.
out	<i>theta_stack,A</i>	sequence of sampled thetas. Size: [model.num_params, nSamples]

Definition at line 45 of file openmp\_methods.h.

References `omp_exception_catcher::Continue`, `omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::run()`.

**8.12.1.18** `template<class Model> void mappel::methods::openmp::simulate_image_stack ( const Model & model, const ParamVecT<Model> & theta_stack, ImageStackT<Model> & image_stack )`

Parallel simulation of images from one or more theta.

This accepts either a single theta and a stack of images, or a stack of thetas and a stack of images.

Use: `model.make_param_stack()` to make a parameter stack of appropriate dimensions for the model Use: `model.make_image_stack()` to make an image stack of appropriate dimensions for the model

### Template Parameters

<i>Model</i>	A concrete subclass of <a href="#">PointEmitterModel</a>
--------------	----------------------------------------------------------

### Parameters

in	<i>model</i>	A <a href="#">PointEmitterModel</a> object.
in	<i>theta_stack</i>	Single theta or a sequence of thetas. Size: [model.num_params, nThetas]
out	<i>image_stack</i>	Sequence of model images generated.

Definition at line 100 of file `openmp_methods.h`.

References `mappel::methods::model_image()`, `omp_exception_catcher::impl_::OMPEExceptionCatcher<_dummy>::rethrow()`, `omp_exception_catcher::impl_::OMPEExceptionCatcher<_dummy>::run()`, and `mappel::methods::simulate_image()`.

## 8.13 omp\_exception\_catcher Namespace Reference

### Namespaces

- [impl\\_](#)

### Typedefs

- using [OMPEExceptionCatcher](#) = [impl\\_::OMPEExceptionCatcher<>](#)

### Enumerations

- enum [Strategy](#) { [Strategy::DoNotTry](#), [Strategy::Continue](#), [Strategy::Abort](#), [Strategy::RethrowFirst](#) }

#### 8.13.1 Typedef Documentation

##### 8.13.1.1 using `omp_exception_catcher::OMPEExceptionCatcher` = `typedef impl_::OMPEExceptionCatcher<>`

A class to run and catch exceptions in parallel code allowing various exception management strategies

Definition at line 114 of file `OMPEExceptionCatcher.h`.

### 8.13.2 Enumeration Type Documentation

#### 8.13.2.1 enum omp\_exception\_catcher::Strategy [strong]

Enumerator

***DoNotTry***  
***Continue***  
***Abort***  
***RethrowFirst***

Definition at line 38 of file OMPEXceptionCatcher.h.

### 8.14 omp\_exception\_catcher::impl\_ Namespace Reference

Classes

- class [OMPEXceptionCatcher](#)

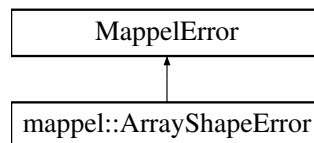
## 9 Class Documentation

### 9.1 mappel::ArrayShapeError Struct Reference

Array is not of the right dimensionality.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/util.h>
```

Inheritance diagram for mappel::ArrayShapeError:



Public Member Functions

- [ArrayShapeError](#) (std::string message)

#### 9.1.1 Detailed Description

Array is not of the right dimensionality.

Definition at line 76 of file util.h.

### 9.1.2 Constructor & Destructor Documentation

#### 9.1.2.1 mappel::ArrayShapeError::ArrayShapeError ( std::string *message* ) [inline]

Definition at line 78 of file util.h.

The documentation for this struct was generated from the following file:

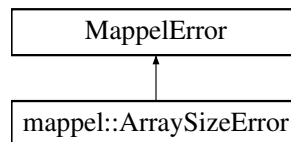
- [util.h](#)

## 9.2 mappel::ArraySizeError Struct Reference

Array is not of the right size.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/util.h>
```

Inheritance diagram for mappel::ArraySizeError:



### Public Member Functions

- [ArraySizeError](#) (std::string message)

#### 9.2.1 Detailed Description

Array is not of the right size.

Definition at line 83 of file util.h.

### 9.2.2 Constructor & Destructor Documentation

#### 9.2.2.1 mappel::ArraySizeError::ArraySizeError ( std::string *message* ) [inline]

Definition at line 85 of file util.h.

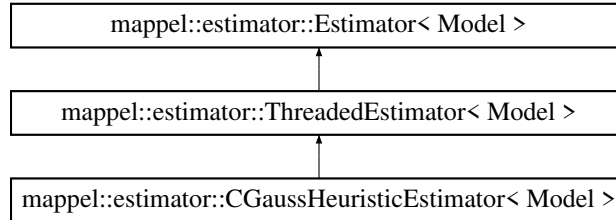
The documentation for this struct was generated from the following file:

- [util.h](#)

## 9.3 mappel::estimator::CGaussHeuristicEstimator&lt; Model &gt; Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::estimator::CGaussHeuristicEstimator< Model >:



## Public Member Functions

- [CGaussHeuristicEstimator](#) (const Model &model)
- [StatsT get\\_stats](#) ()
- [StatsT get\\_debug\\_stats](#) ()
- [std::string name](#) () const
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [\\_stack](#), [MLEDataStack](#) &mle\_data\_stack) override
- void [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [ProfileLikelihoodData](#) &profile) override
- void [estimate\\_profile\\_bounds\\_parallel](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est) override
- void [estimate\\_profile\\_bounds\\_stack](#) (const [ModelDataStackT](#)< Model > &data, [ProfileBoundsDataStack](#) &bounds\_est\_stack) override
- void [clear\\_stats](#) ()
- const Model & [get\\_model](#) ()
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data\_stack, [MLEDataStack](#) &mle\_data\_stack)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, [MLEData](#) &mle\_data)
- void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data)
- double [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed\_idxs, const [ParamT](#)< Model > &fixed\_theta\_init, [StencilT](#)< Model > &theta\_max)
- void [estimate\\_profile\\_bounds](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est)
- void [estimate\\_profile\\_bounds\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &bounds\_est)
- [IdxVecT get\\_exit\\_counts](#) () const

### Protected Member Functions

- void [record\\_exit\\_code](#) (ExitCode code) override
- virtual void [compute\\_estimate\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEDebugData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- virtual double [compute\\_profile\\_estimate](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, const [IdxVecT](#) &fixed\_idx, [StencilT](#)< Model > &max\_stencil)
- virtual void [compute\\_profile\\_bound](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &est, const [VecT](#) &init\_step, [IdxT](#) param\_idx, [IdxT](#) which\_bound)
- virtual void [compute\\_profile\\_bound\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &est)
- void [record\\_walltime](#) (ClockT::time\_point start\_walltime, int num\_estimations)

### Protected Attributes

- int [max\\_threads](#)
- int [num\\_threads](#)
- std::mutex [mtx](#)
- const Model & [model](#)
- int [num\\_estimations](#) = 0
- double [total\\_walltime](#) = 0.
- [IdxVecT](#) [exit\\_counts](#)

### 9.3.1 Detailed Description

```
template<class Model>
class mappel::estimator::CGaussHeuristicEstimator< Model >
```

Definition at line 346 of file estimator.h.

### 9.3.2 Constructor & Destructor Documentation

```
9.3.2.1 template<class Model > mappel::estimator::CGaussHeuristicEstimator< Model
>::CGaussHeuristicEstimator ( const Model & model ) [inline]
```

Definition at line 349 of file estimator.h.

### 9.3.3 Member Function Documentation

```
9.3.3.1 template<class Model > void mappel::estimator::ThreadedEstimator< Model >::clear_stats ( ) [virtual],
[inherited]
```

Run statistics.

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 570 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::clear\\_stats\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::mtx](#), and [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::clear\\_stats\(\)](#).



9.3.3.2 `template<class Model > void mappel::estimator::Estimator< Model >::compute_estimate_debug ( const ModelDataT< Model > & im, const ParamT< Model > & theta_init, MLEDebugData & mle_debug, StencilT< Model > & mle_stencil )` [protected], [virtual], [inherited]

Virtual estimate\_debug interface

Estimators that produce a sequence of results (e.g. IterativeEstimators) can override this dummy debug implementation.

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 285 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDebugData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max_debug()`.

9.3.3.3 `template<class Model > void mappel::estimator::Estimator< Model >::compute_profile_bound ( const ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 309 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

9.3.3.4 `template<class Model > void mappel::estimator::Estimator< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & est )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 318 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`.

9.3.3.5 `template<class Model > double mappel::estimator::Estimator< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxs, StencilT< Model > & max_stencil )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 300 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

**9.3.3.6** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`.

**9.3.3.7** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.

9.3.3.8 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::Estimator< Model >::model`.

9.3.3.9 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at `theta_init`, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

**9.3.3.10** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data )`  
`[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

**9.3.3.11** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack )` `[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max_stack()`, and `mappel::estimator::Estimator< Model >::model`.

9.3.3.12 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack ) [override], [virtual], [inherited]`

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEDataStack::Ndata`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDataStack::obsI`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDataStack::rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDataStack::theta`.

9.3.3.13 `template<class Model> void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

**9.3.3.14** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

**9.3.3.15** `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.

**9.3.3.16** `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, and `mappel::estimator::ProfileBoundsDataStack::initialize_arrays()`.

mappel::estimator::ProfileBoundsData::mle, mappel::estimator::ProfileBoundsDataStack::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsDataStack::Ndata, mappel::estimator::ProfileBoundsDataStack::Nparams\_est, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::MLEData::obsI, mappel::estimator::MLEDataStack::obsI, mappel::print\_text\_image(), mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_ub, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, mappel::estimator::MLEDataStack::rllh, mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, mappel::estimator::ProfileBoundsDataStack::target\_rllh\_delta, mappel::estimator::MLEData::theta, and mappel::estimator::MLEDataStack::theta.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_stack().

**9.3.3.17** `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), and mappel::methods::objective::rllh().

**9.3.3.18** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override], [virtual], [inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::estimator::ProfileLikelihoodData::fixed\_idx, mappel::estimator::ProfileLikelihoodData::fixed\_values, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileLikelihoodData::Nfixed, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::ProfileLikelihoodData::Nvalues, mappel::print\_text\_image(), mappel::estimator::ProfileLikelihoodData::profile\_likelihood, mappel::estimator::ProfileLikelihoodData::profile\_parameters, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), and mappel::estimator::Estimator< Model >::record\_walltime().

**9.3.3.19** `template<class Model> StatsT mappel::estimator::CGaussHeuristicEstimator< Model >::get_debug_stats ( ) [virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 650 of file estimator\_impl.h.

References mappel::estimator::ThreadedEstimator< Model >::get\_stats().

**9.3.3.20** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts ( ) const`  
`[inline], [inherited]`

Run statistics.

Definition at line 274 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.3.3.21** `template<class Model > const Model & mappel::estimator::Estimator< Model >::get_model ( )`  
`[inherited]`

Definition at line 108 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::model`.

**9.3.3.22** `template<class Model > StatsT mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats ( )`  
`[virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 635 of file estimator\_impl.h.

References `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, and `mappel::estimator::Estimator< Model >::num_estimations`.

**9.3.3.23** `template<class Model > std::string mappel::estimator::CGaussHeuristicEstimator< Model >::name ( ) const`  
`[inline], [virtual]`

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 353 of file estimator.h.

**9.3.3.24** `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode`  
`code ) [override], [protected], [virtual], [inherited]`

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.



**9.3.3.25** `template<class Model> void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` [protected], [inherited]

Definition at line 360 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

### 9.3.4 Member Data Documentation

**9.3.4.1** `template<class Model> IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 299 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.3.4.2** `template<class Model> int mappel::estimator::ThreadedEstimator< Model >::max_threads` [protected], [inherited]

Definition at line 324 of file estimator.h.

**9.3.4.3** `template<class Model> const Model& mappel::estimator::Estimator< Model >::model` [protected], [inherited]

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.3.4.4** `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` [protected],  
[inherited]

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.3.4.5** `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected],  
[inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

**9.3.4.6** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected],  
[inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.

**9.3.4.7** `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` [protected],  
[inherited]

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

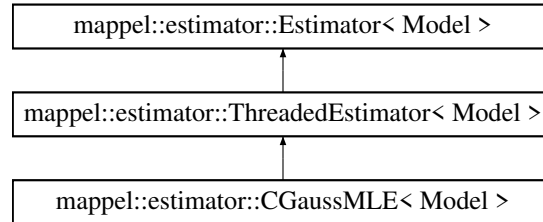
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.4 mappel::estimator::CGaussMLE&lt; Model &gt; Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::estimator::CGaussMLE< Model >:



## Public Member Functions

- [CGaussMLE](#) (const Model &model, int num\_iterations=[DefaultIterations](#))
- [StatsT get\\_stats](#) ()
- [StatsT get\\_debug\\_stats](#) ()
- [std::string name](#) () const
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [\\_stack](#), [MLEDataStack](#) &mle\_data\_stack) override
- void [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [ProfileLikelihoodData](#) &profile) override
- void [estimate\\_profile\\_bounds\\_parallel](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est) override
- void [estimate\\_profile\\_bounds\\_stack](#) (const [ModelDataStackT](#)< Model > &data, [ProfileBoundsDataStack](#) &bounds\_est\_stack) override
- void [clear\\_stats](#) ()
- const Model & [get\\_model](#) ()
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data\_stack, [MLEDataStack](#) &mle\_data\_stack)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, [MLEData](#) &mle\_data)
- void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data)
- double [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed\_idxs, const [ParamT](#)< Model > &fixed\_theta\_init, [StencilT](#)< Model > &theta\_max)
- void [estimate\\_profile\\_bounds](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est)
- void [estimate\\_profile\\_bounds\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &bounds\_est)
- [IdxVecT get\\_exit\\_counts](#) () const

## Static Public Attributes

- static const int [DefaultIterations](#) =50

## Protected Member Functions

- void [record\\_exit\\_code](#) (ExitCode code) override
- virtual double [compute\\_profile\\_estimate](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta, ←  
\_init, const [IdxVecT](#) &fixed\_idx, [StencilT](#)< Model > &max\_stencil)
- virtual void [compute\\_profile\\_bound](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &est, const [VecT](#)  
&init\_step, [IdxT](#) param\_idx, [IdxT](#) which\_bound)
- virtual void [compute\\_profile\\_bound\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &est)
- void [record\\_walltime](#) (ClockT::time\_point start\_walltime, int [num\\_estimations](#))

## Protected Attributes

- int [max\\_threads](#)
- int [num\\_threads](#)
- std::mutex [mtx](#)
- const Model & [model](#)
- int [num\\_estimations](#) = 0
- double [total\\_walltime](#) = 0.
- [IdxVecT](#) [exit\\_counts](#)

### 9.4.1 Detailed Description

```
template<class Model>
class mappel::estimator::CGaussMLE< Model >
```

Definition at line 361 of file estimator.h.

### 9.4.2 Constructor & Destructor Documentation

9.4.2.1 `template<class Model > mappel::estimator::CGaussMLE< Model >::CGaussMLE ( const Model & model, int num_iterations = DefaultIterations ) [inline]`

Definition at line 366 of file estimator.h.

## 9.4.3 Member Function Documentation

9.4.3.1 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::clear_stats ( ) [virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 570 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::clear\\_stats\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::mtx](#), and [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::clear\\_stats\(\)](#).

9.4.3.2 `template<class Model> void mappel::estimator::Estimator< Model >::compute_profile_bound ( const ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound ) [protected], [virtual], [inherited]`

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 309 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::model](#).

Referenced by [mappel::estimator::Estimator< Model >::estimate\\_profile\\_bounds\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_parallel\(\)](#), and [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_stack\(\)](#).

9.4.3.3 `template<class Model> void mappel::estimator::Estimator< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & est ) [protected], [virtual], [inherited]`

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 318 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::model](#).

Referenced by [mappel::estimator::Estimator< Model >::estimate\\_profile\\_bounds\\_debug\(\)](#).

9.4.3.4 `template<class Model> double mappel::estimator::Estimator< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxs, StencilT< Model > & max_stencil ) [protected], [virtual], [inherited]`

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 300 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::model](#).

Referenced by [mappel::estimator::Estimator< Model >::estimate\\_profile\\_max\(\)](#), and [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_max\(\)](#).

**9.4.3.5** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`.

**9.4.3.6** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.

9.4.3.7 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::Estimator< Model >::model`.

9.4.3.8 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at `theta_init`, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

**9.4.3.9** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data )`  
`[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

**9.4.3.10** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack )` `[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max_stack()`, and `mappel::estimator::Estimator< Model >::model`.



9.4.3.11 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack ) [override], [virtual], [inherited]`

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEDataStack::Ndata`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDataStack::obsI`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDataStack::rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDataStack::theta`.

9.4.3.12 `template<class Model> void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

**9.4.3.13** `template<class Model> void mappel::estimator::Estimator< Model>::estimate_profile_bounds_debug ( const ModelDataT< Model> & data, ProfileBoundsDebugData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model>::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model>::record_exit_code()`, `mappel::estimator::Estimator< Model>::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

**9.4.3.14** `template<class Model> void mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_parallel ( const ModelDataT< Model> & data, ProfileBoundsData & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model>](#).

Definition at line 464 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model>::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model>::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model>::num_threads`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`, `mappel::estimator::Estimator< Model>::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.

**9.4.3.15** `template<class Model> void mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_stack ( const ModelDataStackT< Model> & data_stack, ProfileBoundsDataStack & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model>](#).

Definition at line 500 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model>::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, and `mappel::estimator::ProfileBoundsDataStack::initialize_arrays()`.

mappel::estimator::ProfileBoundsData::mle, mappel::estimator::ProfileBoundsDataStack::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsDataStack::Ndata, mappel::estimator::ProfileBoundsDataStack::Nparams\_est, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::MLEData::obsI, mappel::estimator::MLEDataStack::obsI, mappel::print\_text\_image(), mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_ub, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, mappel::estimator::MLEDataStack::rllh, mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, mappel::estimator::ProfileBoundsDataStack::target\_rllh\_delta, mappel::estimator::MLEData::theta, and mappel::estimator::MLEDataStack::theta.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_stack().

**9.4.3.16** `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), and mappel::methods::objective::rllh().

**9.4.3.17** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override], [virtual], [inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::estimator::ProfileLikelihoodData::fixed\_idx, mappel::estimator::ProfileLikelihoodData::fixed\_values, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileLikelihoodData::Nfixed, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::ProfileLikelihoodData::Nvalues, mappel::print\_text\_image(), mappel::estimator::ProfileLikelihoodData::profile\_likelihood, mappel::estimator::ProfileLikelihoodData::profile\_parameters, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), and mappel::estimator::Estimator< Model >::record\_walltime().

**9.4.3.18** `template<class Model> StatsT mappel::estimator::CGaussMLE< Model>::get_debug_stats ( )`  
`[virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model>](#).

Definition at line 675 of file `estimator_impl.h`.

References `mappel::cgauss_compute_estimate()`, `mappel::cgauss_compute_estimate_debug()`, `mappel::cgauss_heuristic_compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::ThreadedEstimator< Model>::get_stats()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::methods::objective::openmp::rllh_stack()`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::Success`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDebugData::theta`.

**9.4.3.19** `template<class Model> IdxVecT mappel::estimator::Estimator< Model>::get_exit_counts ( ) const`  
`[inline], [inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.4.3.20** `template<class Model> const Model & mappel::estimator::Estimator< Model>::get_model ( )`  
`[inherited]`

Definition at line 108 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model>::model`.

**9.4.3.21** `template<class Model> StatsT mappel::estimator::CGaussMLE< Model>::get_stats ( )` `[virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model>](#).

Definition at line 660 of file `estimator_impl.h`.

References `mappel::estimator::ThreadedEstimator< Model>::get_stats()`, `mappel::estimator::ThreadedEstimator< Model>::mtx`, and `mappel::estimator::Estimator< Model>::num_estimations`.

**9.4.3.22** `template<class Model> std::string mappel::estimator::CGaussMLE< Model>::name ( ) const` `[inline], [virtual]`

Implements [mappel::estimator::Estimator< Model>](#).

Definition at line 371 of file `estimator.h`.

**9.4.3.23** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code )` `[override]`, `[protected]`, `[virtual]`, `[inherited]`

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::Success`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.4.3.24** `template<class Model> void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` `[protected]`, `[inherited]`

Definition at line 360 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

## 9.4.4 Member Data Documentation

**9.4.4.1** `template<class Model> const int mappel::estimator::CGaussMLE< Model >::DefaultIterations = 50` `[static]`

Definition at line 364 of file `estimator.h`.

**9.4.4.2** `template<class Model> IdxVecT mappel::estimator::Estimator< Model >::exit_counts` `[protected]`, `[inherited]`

Definition at line 299 of file `estimator.h`.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.4.4.3** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::max_threads` [protected],  
[inherited]

Definition at line 324 of file estimator.h.

**9.4.4.4** `template<class Model > const Model& mappel::estimator::Estimator< Model >::model` [protected],  
[inherited]

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.4.4.5** `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` [protected],  
[inherited]

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.4.4.6** `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected],  
[inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

9.4.4.7 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.

9.4.4.8 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` [protected], [inherited]

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

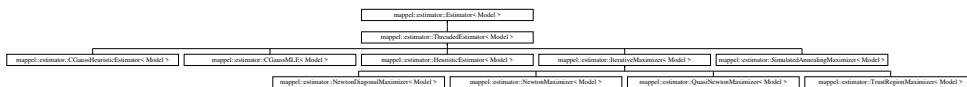
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.5 mappel::estimator::Estimator< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::estimator::Estimator< Model >`:



### Public Member Functions

- [Estimator](#) (const Model &\_model)
- virtual [~Estimator](#) ()
- virtual std::string [name](#) () const =0
- const Model & [get\\_model](#) ()
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, [MLEData](#) &mle\_data)

- void `estimate_max_debug` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDebugData` &mle\_data, `StencilT`< `Model` > &mle\_stencil)
- void `estimate_max_debug` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDebugData` &mle\_data)
- virtual void `estimate_max_stack` (const `ModelDataStackT`< `Model` > &data\_stack, const `ParamVecT`< `Model` > &theta\_init\_stack, `MLEDDataStack` &mle\_data\_stack)=0
- void `estimate_max_stack` (const `ModelDataStackT`< `Model` > &data\_stack, `MLEDDataStack` &mle\_data\_stack)
- double `estimate_profile_max` (const `ModelDataT`< `Model` > &data, const `IdxVecT` &fixed\_idx, const `ParamT`< `Model` > &fixed\_theta\_init, `StencilT`< `Model` > &theta\_max)
- virtual void `estimate_profile_max` (const `ModelDataT`< `Model` > &data, const `ParamVecT`< `Model` > &fixed\_theta\_init, `ProfileLikelihoodData` &profile)=0
- void `estimate_profile_bounds` (const `ModelDataT`< `Model` > &data, `ProfileBoundsData` &bounds\_est)
- virtual void `estimate_profile_bounds_parallel` (const `ModelDataT`< `Model` > &data, `ProfileBoundsData` &bounds\_est)=0
- void `estimate_profile_bounds_debug` (const `ModelDataT`< `Model` > &data, `ProfileBoundsDebugData` &bounds\_est)
- virtual void `estimate_profile_bounds_stack` (const `ModelDataStackT`< `Model` > &data\_stack, `ProfileBoundsDataStack` &bounds\_est)=0
- virtual `StatsT` `get_stats` ()
- virtual `StatsT` `get_debug_stats` ()=0
- virtual void `clear_stats` ()
- `IdxVecT` `get_exit_counts` () const

### Protected Member Functions

- virtual void `compute_estimate` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDData` &mle\_data, `StencilT`< `Model` > &mle\_stencil)=0
- virtual void `compute_estimate_debug` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDDebugData` &mle\_data, `StencilT`< `Model` > &mle\_stencil)
- virtual double `compute_profile_estimate` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, const `IdxVecT` &fixed\_idx, `StencilT`< `Model` > &max\_stencil)
- virtual void `compute_profile_bound` (const `ModelDataT`< `Model` > &data, `ProfileBoundsData` &est, const `VecT` &init\_step, `IdxT` param\_idx, `IdxT` which\_bound)
- virtual void `compute_profile_bound_debug` (const `ModelDataT`< `Model` > &data, `ProfileBoundsDebugData` &est)
- void `record_walltime` (ClockT::time\_point start\_walltime, int num\_estimations)
- virtual void `record_exit_code` (ExitCode code)=0

### Protected Attributes

- const `Model` & `model`
- int `num_estimations` = 0
- double `total_walltime` = 0.
- `IdxVecT` `exit_counts`



## Friends

- `template<class T >`  
`std::ostream & operator<< (std::ostream &out, Estimator< T > &estimator)`

## 9.5.1 Detailed Description

`template<class Model>`  
`class mappel::estimator::Estimator< Model >`

[Estimator](#) base class defines the interface for estimator interactions designed to unify the [ThreadedEstimator](#) with future GPUestimator types under a single API.

Design notes: Templated on the model type to allow for direct function call for models through the [mappel::methods](#) namespace templated model methods.

Definition at line 194 of file estimator.h.

## 9.5.2 Constructor &amp; Destructor Documentation

9.5.2.1 `template<class Model > mappel::estimator::Estimator< Model >::Estimator ( const Model &_model )`

Definition at line 102 of file estimator\_impl.h.

9.5.2.2 `template<class Model > virtual mappel::estimator::Estimator< Model >::~Estimator ( ) [inline],`  
`[virtual]`

Definition at line 197 of file estimator.h.

References [mappel::methods::estimate\\_max\(\)](#), [mappel::methods::debug::estimate\\_max\\_debug\(\)](#), and [mappel::methods::openmp::estimate\\_max\\_stack\(\)](#).

## 9.5.3 Member Function Documentation

9.5.3.1 `template<class Model > void mappel::estimator::Estimator< Model >::clear_stats ( ) [virtual]`

Run statistics.

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#), and [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 343 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::exit\\_counts](#), [mappel::estimator::Estimator< Model >::num\\_← estimations](#), and [mappel::estimator::Estimator< Model >::total\\_walltime](#).

Referenced by [mappel::estimator::ThreadedEstimator< Model >::clear\\_stats\(\)](#).

9.5.3.2 `template<class Model > virtual void mappel::estimator::Estimator< Model >::compute_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil ) [protected], [pure virtual]`

Implemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Referenced by `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`.

9.5.3.3 `template<class Model > void mappel::estimator::Estimator< Model >::compute_estimate_debug ( const ModelDataT< Model > & im, const ParamT< Model > & theta_init, MLEDebugData & mle_debug, StencilT< Model > & mle_stencil ) [protected], [virtual]`

Virtual `estimate_debug` interface

Estimators that produce a sequence of results (e.g. `IterativeEstimators`) can override this dummy debug implementation.

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 285 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDebugData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max_debug()`.

9.5.3.4 `template<class Model > void mappel::estimator::Estimator< Model >::compute_profile_bound ( const ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound ) [protected], [virtual]`

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 309 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

9.5.3.5 `template<class Model > void mappel::estimator::Estimator< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & est ) [protected], [virtual]`

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 318 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`.

9.5.3.6 `template<class Model > double mappel::estimator::Estimator< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxes, StencilT< Model > & max_stencil ) [protected], [virtual]`

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 300 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::model](#).

Referenced by [mappel::estimator::Estimator< Model >::estimate\\_profile\\_max\(\)](#), and [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_max\(\)](#).

9.5.3.7 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil )`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::compute\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::MLEData::obsI](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::Estimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), [mappel::estimator::MLEData::rllh](#), and [mappel::estimator::MLEData::theta](#).

Referenced by [mappel::estimator::Estimator< Model >::estimate\\_max\(\)](#).

9.5.3.8 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data )`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.

**9.5.3.9** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data )`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::Estimator< Model >::model`.

**9.5.3.10** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil )`

Debug estimation for a single data starting at `theta_init`, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

**9.5.3.11** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data )`

Maximum likelihood point estimators Estimate for a single data starting at *theta\_init*, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with *theta\_init*, *theta\_init* is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of *theta\_init* will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

**9.5.3.12** `template<class Model > virtual void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack ) [pure virtual]`

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with *theta\_init*, *theta\_init* is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of *theta\_init* will not be modified in the initialization process.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implemented in [mappel::estimator::ThreadedEstimator< Model >](#).

Referenced by [mappel::estimator::Estimator< Model >::estimate\\_max\\_stack\(\)](#).

**9.5.3.13** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack )`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::estimate\\_max\\_stack\(\)](#), and [mappel::estimator::Estimator< Model >::model](#).

**9.5.3.14** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est )`

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_bound\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::ProfileBoundsData::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsData::initialize\\_arrays\(\)](#), [mappel::estimator::ProfileBoundsData::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsData::Nparams\\_est](#), [mappel::estimator::MLEData::obs](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ProfileBoundsData::profile\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsData::profile\\_ub](#), [mappel::estimator::Estimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), [mappel::estimator::subroutine::solve\\_profile\\_initial\\_step\(\)](#), [mappel::estimator::ProfileBoundsData::target\\_rllh\\_delta](#), and [mappel::estimator::MLEData::theta](#).

Referenced by [mappel::methods::error\\_bounds\\_profile\\_likelihood\(\)](#).

9.5.3.15 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est )`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

9.5.3.16 `template<class Model > virtual void mappel::estimator::Estimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [pure virtual]`

Profile likelihood bounds computations with VM algorithm

Implemented in `mappel::estimator::ThreadedEstimator< Model >`.

9.5.3.17 `template<class Model > virtual void mappel::estimator::Estimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est ) [pure virtual]`

Profile likelihood bounds computations with VM algorithm

Implemented in `mappel::estimator::ThreadedEstimator< Model >`.

9.5.3.18 `template<class Model > double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max )`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::Error`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

9.5.3.19 `template<class Model > virtual void mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [pure virtual]`

Profile likelihood estimation methods

Implemented in `mappel::estimator::ThreadedEstimator< Model >`.

**9.5.3.20** `template<class Model > virtual StatsT mappel::estimator::Estimator< Model >::get_debug_stats ( ) [pure virtual]`

Run statistics.

Implemented in [mappel::estimator::IterativeMaximizer< Model >](#), [mappel::estimator::SimulatedAnnealingMaximizer< Model >](#), [mappel::estimator::CGaussMLE< Model >](#), [mappel::estimator::CGaussHeuristicEstimator< Model >](#), [mappel::estimator::HeuristicEstimator< Model >](#), and [mappel::estimator::ThreadedEstimator< Model >](#).

**9.5.3.21** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts ( ) const [inline]`

Run statistics.

Definition at line 274 of file estimator.h.

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

**9.5.3.22** `template<class Model > const Model & mappel::estimator::Estimator< Model >::get_model ( )`

Definition at line 108 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::model](#).

**9.5.3.23** `template<class Model > StatsT mappel::estimator::Estimator< Model >::get_stats ( ) [virtual]`

Run statistics.

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#), [mappel::estimator::SimulatedAnnealingMaximizer< Model >](#), [mappel::estimator::CGaussMLE< Model >](#), [mappel::estimator::CGaussHeuristicEstimator< Model >](#), [mappel::estimator::HeuristicEstimator< Model >](#), and [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 326 of file estimator\_impl.h.

References [mappel::estimator::Error](#), [mappel::estimator::Estimator< Model >::exit\\_counts](#), [mappel::estimator::GradRatio](#), [mappel::estimator::MaxBacktracks](#), [mappel::estimator::MaxIter](#), [mappel::estimator::Estimator< Model >::num\\_estimations](#), [mappel::estimator::StepSize](#), [mappel::estimator::Success](#), [mappel::estimator::Estimator< Model >::total\\_walltime](#), and [mappel::estimator::TrustRegionRadius](#).

Referenced by [mappel::estimator::ThreadedEstimator< Model >::get\\_stats\(\)](#).

**9.5.3.24** `template<class Model > virtual std::string mappel::estimator::Estimator< Model >::name ( ) const [pure virtual]`

Implemented in [mappel::estimator::TrustRegionMaximizer< Model >](#), [mappel::estimator::QuasiNewtonMaximizer< Model >](#), [mappel::estimator::NewtonMaximizer< Model >](#), [mappel::estimator::NewtonDiagonalMaximizer< Model >](#), [mappel::estimator::SimulatedAnnealingMaximizer< Model >](#), [mappel::estimator::CGaussMLE< Model >](#), [mappel::estimator::CGaussHeuristicEstimator< Model >](#), and [mappel::estimator::HeuristicEstimator< Model >](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::solve\\_profile\\_bound\(\)](#).



9.5.3.25 `template<class Model > virtual void mappel::estimator::Estimator< Model >::record_exit_code ( ExitCode code ) [protected],[pure virtual]`

Implemented in [mappel::estimator::ThreadedEstimator< Model >](#).

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, and `mappel::estimator::Estimator< Model >::estimate_profile_max()`.

9.5.3.26 `template<class Model > void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations ) [protected]`

Definition at line 360 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

## 9.5.4 Friends And Related Function Documentation

9.5.4.1 `template<class Model > template<class T > std::ostream& operator<< ( std::ostream & out, Estimator< T > & estimator ) [friend]`

## 9.5.5 Member Data Documentation

9.5.5.1 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts [protected]`

Definition at line 299 of file `estimator.h`.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

### 9.5.5.2 `template<class Model > const Model& mappel::estimator::Estimator< Model >::model` [protected]

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

### 9.5.5.3 `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

### 9.5.5.4 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` [protected]

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

The documentation for this class was generated from the following files:

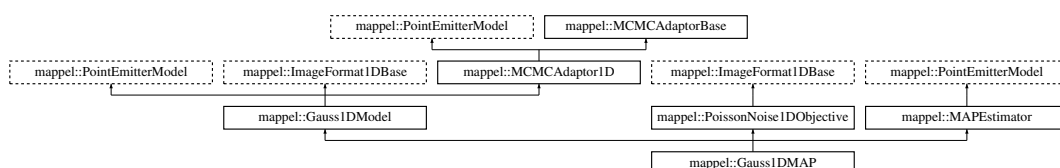
- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.6 mappel::Gauss1DMP Class Reference

A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DMP.h>
```

Inheritance diagram for `mappel::Gauss1DMP`:



## Public Types

- using `StencilVecT` = `std::vector< Stencil >`
- using `ParamT` = `arma::vec`
- using `ParamVecT` = `arma::mat`
- using `ImageCoordT` = `uint32_t`
- using `ImagePixelT` = `double`
- template<class `CoordT` >  
using `ImageSizeShapeT` = `CoordT`
- template<class `CoordT` >  
using `ImageSizeVecShapeT` = `arma::Col< CoordT >`
- using `ImageSizeT` = `ImageSizeShapeT< ImageCoordT >`
- using `ImageSizeVecT` = `ImageSizeVecShapeT< ImageCoordT >`
- template<class `PixelT` >  
using `ImageShapeT` = `arma::Col< PixelT >`
- template<class `PixelT` >  
using `ImageStackShapeT` = `arma::Mat< PixelT >`
- using `ImageT` = `ImageShapeT< ImagePixelT >`
- using `ImageStackT` = `ImageStackShapeT< ImagePixelT >`
- using `ModelDataT` = `ImageT`
- using `ModelDataStackT` = `ImageStackT`

## Public Member Functions

- `Gauss1DMAP` (`arma::Col< ImageCoordT > size`, `VecT psf_sigma`, `const std::string &prior_type=DefaultPriorType`)
- `Gauss1DMAP` (`ImageSizeT size`, `double psf_sigma`, `const std::string &prior_type=DefaultPriorType`)
- `Gauss1DMAP` (`ImageSizeT size`, `double psf_sigma`, `CompositeDist &&prior`)
- `Gauss1DMAP` (`ImageSizeT size`, `double psf_sigma`, `const CompositeDist &prior`)
- `Gauss1DMAP` (`const Gauss1DMAP &o`)
- `Gauss1DMAP & operator=` (`const Gauss1DMAP &o`)
- `Gauss1DMAP` (`Gauss1DMAP &&o`)
- `Gauss1DMAP & operator=` (`Gauss1DMAP &&o`)
- `double get_psf_sigma` () `const`
- `double get_psf_sigma` (`IdxT idx`) `const`
- `void set_psf_sigma` (`double new_psf_sigma`)
- `void set_psf_sigma` (`const VecT &new_psf_sigma`)
- `StatsT get_stats` () `const`
- `Stencil make_stencil` (`const ParamT &theta`, `bool compute_derivatives=true`) `const`  
*Make a new Model::Stencil object at theta.*
- `double pixel_model_value` (`IdxT i`, `const Stencil &s`) `const`
- `void pixel_grad` (`IdxT i`, `const Stencil &s`, `ParamT &pgrad`) `const`
- `void pixel_grad2` (`IdxT i`, `const Stencil &s`, `ParamT &pgrad2`) `const`
- `void pixel_hess` (`IdxT i`, `const Stencil &s`, `MatT &hess`) `const`
- `void pixel_hess_update` (`IdxT i`, `const Stencil &s`, `double dm_ratio_m1`, `double dmm_ratio`, `ParamT &grad`, `MatT &hess`) `const`  
*pixel derivative inner loop calculations.*
- `Stencil initial_theta_estimate` (`const ImageT &im`) `const`  
*Fast, heuristic estimate of initial theta.*
- `Stencil initial_theta_estimate` (`const ImageT &im`, `const ParamT &theta_init`) `const`

- `IdxT get_num_params () const`
- `void check_param_shape (const ParamT &theta) const`
- `void check_param_shape (const ParamVecT &theta) const`
- `void check_psf_sigma (double psf_sigma) const`
- `void check_psf_sigma (const VecT &psf_sigma) const`
- `ParamT make_param () const`
- `template<class FillT >`  
`ParamT make_param (FillT fill) const`
- `ParamVecT make_param_stack (IdxT n) const`
- `template<class FillT >`  
`ParamVecT make_param_stack (IdxT n, FillT fill) const`
- `MatT make_param_mat () const`
- `template<class FillT >`  
`MatT make_param_mat (FillT fill) const`
- `CubeT make_param_mat_stack (IdxT n) const`
- `template<class FillT >`  
`CubeT make_param_mat_stack (IdxT n, FillT fill) const`
- `CompositeDist & get_prior ()`
- `const CompositeDist & get_prior () const`
- `void set_prior (CompositeDist &&prior_)`
- `void set_prior (const CompositeDist &prior_)`
- `IdxT get_num_hyperparams () const`
- `void set_hyperparams (const VecT &hyperparams)`
- `VecT get_hyperparams () const`
- `bool has_hyperparam (const std::string &name) const`
- `double get_hyperparam_value (const std::string &name) const`
- `int get_hyperparam_index (const std::string &name) const`
- `void set_hyperparam_value (const std::string &name, double value)`
- `void rename_hyperparam (const std::string &old_name, const std::string &new_name)`
- `StringVecT get_param_names () const`
- `void set_param_names (const StringVecT &desc)`
- `StringVecT get_hyperparam_names () const`
- `void set_hyperparam_names (const StringVecT &desc)`
- `template<class RngT >`  
`ParamT sample_prior (RngT &rng) const`
- `ParamT sample_prior () const`
- `void set_bounds (const ParamT &lbound, const ParamT &ubound)`
- `void set_lbound (const ParamT &lbound)`
- `void set_ubound (const ParamT &ubound)`
- `const ParamT & get_lbound () const`
- `const ParamT & get_ubound () const`
- `bool theta_in_bounds (const ParamT &theta) const`
- `void bound_theta (ParamT &theta, double epsilon=bounds_epsilon) const`
- `ParamT bounded_theta (const ParamT &theta, double epsilon=bounds_epsilon) const`
- `ParamT reflected_theta (const ParamT &theta) const`
- `BoolVecT theta_stack_in_bounds (const ParamVecT &theta) const`
- `ParamVecT bounded_theta_stack (const ParamVecT &theta, double epsilon=bounds_epsilon) const`
- `ParamVecT reflected_theta_stack (const ParamVecT &theta) const`
- `ImageT make_image () const`
- `ImageStackT make_image_stack (ImageCoordT n) const`
- `ImageCoordT get_size_image_stack (const ImageStackT &stack) const`

- [ImageT get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- template<class ImT >  
void [set\\_image\\_in\\_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const ImT &im) const
- [ImageSizeT get\\_size](#) () const
- [ImageCoordT get\\_size](#) ([IdxT](#) idx) const
- [ImageCoordT get\\_num\\_pixels](#) () const
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- void [set\\_size](#) (const arma::Col< [ImageCoordT](#) > &sz)
- void [check\\_image\\_shape](#) (const [ImageT](#) &im) const  
*Check the shape of a single images is correct for model size.*
- void [check\\_image\\_shape](#) (const [ImageStackT](#) &ims) const  
*Check the shape of a stack of images is correct for model size.*
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, const [IdxVecT](#) &fixed\_parameters\_↵ mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)
- void [set\\_background\\_mcmc\\_sampling](#) (double eta\_bg=-1)
- void [set\\_mcmc\\_sigma\\_scale](#) (double scale)
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- [IdxT get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Member Functions

- static CompositeDist [make\\_default\\_prior](#) ([IdxT](#) size, const std::string &prior\_type)
- static CompositeDist [make\\_default\\_prior\\_beta\\_position](#) ([IdxT](#) size)
- static CompositeDist [make\\_default\\_prior\\_normal\\_position](#) ([IdxT](#) size)
- static CompositeDist [make\\_prior\\_beta\\_position](#) ([IdxT](#) size, double beta\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static CompositeDist [make\\_prior\\_normal\\_position](#) ([IdxT](#) size, double sigma\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static prior\_hessian::TruncatedNormalDist [make\\_prior\\_component\\_position\\_normal](#) ([IdxT](#) size, double pos\_↵ sigma=default\_sigma\_pos)
- static prior\_hessian::ScaledSymmetricBetaDist [make\\_prior\\_component\\_position\\_beta](#) ([IdxT](#) size, double pos\_↵ beta=default\_beta\_pos)
- static prior\_hessian::TruncatedGammaDist [make\\_prior\\_component\\_intensity](#) (double mean=default\_mean\_↵ l, double kappa=default\_intensity\_kappa)
- static prior\_hessian::TruncatedParetoDist [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵ sigma, double alpha=default\_alpha\_sigma)
- static void [set\\_rng\\_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)

*Check the size argument for the model.*

### Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_I](#) = 300
- static const double [default\\_max\\_I](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) = 1
- static const [ImageCoordT](#) [global\\_min\\_size](#) = 3
- static const [ImageCoordT](#) [global\\_max\\_size](#) = 512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5
- static const std::vector< std::string > [estimator\\_names](#)

### Protected Member Functions

- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) [num\\_phases](#))

### Protected Attributes

- double [psf\\_sigma](#)
- CompositeDist [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta\\_x](#) =0
- double [eta\\_I](#) =0
- double [eta\\_bg](#) =0
- [IdxT](#) [num\\_phases](#)
- double [sigma\\_scale](#)

### 9.6.1 Detailed Description

A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.

Model: [Gauss1DModel](#) - 1D Gaussian PSF with fixed PSF sigma Objective: [PoissonNoise1DObjective](#) - Poisson noise model for 1D Estimator: MAPstimator - Maximum a-posteriori estimator

Definition at line 23 of file Gauss1DMAP.h.

### 9.6.2 Member Typedef Documentation

#### 9.6.2.1 using mappel::ImageFormat1DBase::ImageCoordT = uint32\_t [inherited]

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

#### 9.6.2.2 using mappel::ImageFormat1DBase::ImagePixelT = double [inherited]

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

#### 9.6.2.3 template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT> [inherited]

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

#### 9.6.2.4 template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT [inherited]

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

#### 9.6.2.5 using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT> [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

#### 9.6.2.6 template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT> [inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

#### 9.6.2.7 using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT> [inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

**9.6.2.8** `template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

**9.6.2.9** `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

**9.6.2.10** `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

**9.6.2.11** `using mappel::PoissonNoise1DObjective::ModelDataStackT = ImageStackT` `[inherited]`

Objective function data stack type: 1D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 26 of file PoissonNoise1DObjective.h.

**9.6.2.12** `using mappel::PoissonNoise1DObjective::ModelDataT = ImageT` `[inherited]`

Objective function data type: 1D double precision image, gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise1DObjective.h.

**9.6.2.13** `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

**9.6.2.14** `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

**9.6.2.15** `using mappel::Gauss1DModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 49 of file Gauss1DModel.h.



## 9.6.3 Constructor &amp; Destructor Documentation

9.6.3.1 **mappel::Gauss1DMAP::Gauss1DMAP** ( arma::Col< ImageCoordT > *size*, VecT *psf\_sigma*, const std::string & *prior\_type* = DefaultPriorType )

Definition at line 11 of file Gauss1DMAP.cpp.

9.6.3.2 **mappel::Gauss1DMAP::Gauss1DMAP** ( ImageSizeT *size*, double *psf\_sigma*, const std::string & *prior\_type* = DefaultPriorType )

Definition at line 15 of file Gauss1DMAP.cpp.

9.6.3.3 **mappel::Gauss1DMAP::Gauss1DMAP** ( ImageSizeT *size*, double *psf\_sigma*, CompositeDist && *prior* )

Definition at line 19 of file Gauss1DMAP.cpp.

9.6.3.4 **mappel::Gauss1DMAP::Gauss1DMAP** ( ImageSizeT *size*, double *psf\_sigma*, const CompositeDist & *prior* )

Definition at line 27 of file Gauss1DMAP.cpp.

9.6.3.5 **mappel::Gauss1DMAP::Gauss1DMAP** ( const Gauss1DMAP & *o* )

Definition at line 35 of file Gauss1DMAP.cpp.

9.6.3.6 **mappel::Gauss1DMAP::Gauss1DMAP** ( Gauss1DMAP && *o* )

Definition at line 43 of file Gauss1DMAP.cpp.

## 9.6.4 Member Function Documentation

9.6.4.1 **void mappel::PointEmitterModel::bound\_theta** ( ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const  
[*inherited*]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

9.6.4.2 **PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta** ( const ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const [*inherited*]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

**9.6.4.3** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` `[inherited]`

Definition at line 306 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

**9.6.4.4** `void ImageFormat1DBase::check_image_shape ( const ImageT & im ) const` `[inherited]`

Check the shape of a single images is correct for model size.

Definition at line 59 of file `ImageFormat1DBase.cpp`.

References `mappel::ImageFormat1DBase::size`.

**9.6.4.5** `void ImageFormat1DBase::check_image_shape ( const ImageStackT & ims ) const` `[inherited]`

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file `ImageFormat1DBase.cpp`.

References `mappel::ImageFormat1DBase::size`.

**9.6.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` `[inherited]`

Definition at line 167 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.6.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 176 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::num_params`.

**9.6.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` `[inherited]`

Definition at line 185 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

9.6.4.9 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

9.6.4.10 `void ImageFormat1DBase::check_size ( const ImageSizeT & size_ )` [static],[inherited]

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::global_max_size`, and `mappel::ImageFormat1DBase::global_min_size`.

Referenced by `mappel::ImageFormat1DBase::ImageFormat1DBase()`, and `mappel::ImageFormat1DBase::set_size()`.

9.6.4.11 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline],[inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.6.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` [inline],[inherited]

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.6.4.13 `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.6.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` [inline],[inherited]

Definition at line 224 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.6.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` [inline],[inherited]

Definition at line 108 of file ImageFormat1DBase.h.

**9.6.4.16** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.6.4.17** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

**9.6.4.18** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

**9.6.4.19** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline]`, `[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

**9.6.4.20** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline]`, `[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.6.4.21** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_num_pixels ( ) const` `[inline]`,  
`[inherited]`

Definition at line 82 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.6.4.22** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const` `[inline]`, `[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.6.4.23 CompositeDist & mappel::PointEmitterModel::get\_prior ( ) [inline],[inherited]**

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.6.4.24 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const [inline],[inherited]**

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.6.4.25 double mappel::Gauss1DModel::get\_psf\_sigma ( ) const [inline],[inherited]**

Definition at line 127 of file Gauss1DModel.h.

References mappel::Gauss1DModel::psf\_sigma.

Referenced by mappel::Gauss1DModel::get\_stats().

**9.6.4.26 double mappel::Gauss1DModel::get\_psf\_sigma ( IdxT idx ) const [inherited]**

Definition at line 131 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::psf\_sigma.

**9.6.4.27 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( ) [static],[inherited]**

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.6.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( ) [static],[inherited]**

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.6.4.29 ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get\_size ( ) const [inline],[inherited]**

Definition at line 71 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.6.4.30** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size ( IdxT idx ) const` `[inherited]`

Definition at line 20 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::size`.

**9.6.4.31** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack ( const ImageStackT & stack ) const` `[inline],[inherited]`

Definition at line 101 of file ImageFormat1DBase.h.

**9.6.4.32** `StatsT mappel::Gauss1DModel::get_stats ( ) const` `[inherited]`

Definition at line 178 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::get_psf_sigma()`, `mappel::MCMCAdaptor1D::get_stats()`, `mappel::ImageFormat1DBase::get_stats()`, and `mappel::PointEmitterModel::get_stats()`.

**9.6.4.33** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound ( ) const` `[inline],[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DsxyModel::get_max_sigma_ratio()`, `mappel::Gauss2DsModel::get_max_sigma_ratio()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

**9.6.4.34** `bool mappel::PointEmitterModel::has_hyperparam ( const std::string & name ) const` `[inline],[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.6.4.35** `Gauss1DModel::Stencil mappel::Gauss1DModel::initial_theta_estimate ( const ImageT & im ) const` `[inline],[inherited]`

Fast, heuristic estimate of initial theta.

Definition at line 169 of file Gauss1DModel.h.

References `mappel::PointEmitterModel::make_param()`, and `mappel::Gauss1DModel::Stencil::theta`.

**9.6.4.36** `Gauss1DModel::Stencil mappel::Gauss1DModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init ) const` `[inherited]`

Definition at line 207 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::Stencil::bg()`, `mappel::Gauss1DModel::Stencil::l()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::PointEmitterModel::num_params`, and `mappel::ImageFormat1DBase::size`.

**9.6.4.37 CompositeDist** mappel::Gauss1DModel::make\_default\_prior ( *IdxT size*, const std::string & *prior\_type* ) [static],  
[inherited]

Definition at line 59 of file Gauss1DModel.cpp.

References mappel::istarts\_with(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), and mappel::Gauss1DModel::make\_default\_prior\_normal\_position().

**9.6.4.38 CompositeDist** mappel::Gauss1DModel::make\_default\_prior\_beta\_position ( *IdxT size* ) [static],  
[inherited]

Definition at line 80 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component←\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_beta().

Referenced by mappel::Gauss1DModel::make\_default\_prior().

**9.6.4.39 CompositeDist** mappel::Gauss1DModel::make\_default\_prior\_normal\_position ( *IdxT size* ) [static],  
[inherited]

Definition at line 90 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component←\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_normal().

Referenced by mappel::Gauss1DModel::make\_default\_prior().

**9.6.4.40 ImageFormat1DBase::ImageT** ImageFormat1DBase::make\_image ( ) const [inline], [inherited]

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.6.4.41 ImageFormat1DBase::ImageStackT** ImageFormat1DBase::make\_image\_stack ( *ImageCoordT n* ) const  
[inline], [inherited]

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.6.4.42 PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline], [inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

**9.6.4.43** `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const`  
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.6.4.44** `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.6.4.45** `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.6.4.46** `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.6.4.47** `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`  
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.6.4.48** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

**9.6.4.49** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.



9.6.4.50 **CompositeDist** mappel::Gauss1DModel::make\_prior\_beta\_position ( *IdxT size*, double *beta\_xpos*, double *mean\_l*, double *kappa\_l*, double *mean\_bg*, double *kappa\_bg* ) [static],[inherited]

Definition at line 101 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_beta().

Referenced by mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator().

9.6.4.51 **prior\_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make\_prior\_component\_intensity ( double *mean = default\_mean\_l*, double *kappa = default\_intensity\_kappa* ) [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_l.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.6.4.52 **prior\_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make\_prior\_component\_position\_beta ( *IdxT size*, double *pos\_beta = default\_beta\_pos* ) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), and mappel::Gauss2DsModel::make\_prior\_beta\_position().

9.6.4.53 **prior\_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make\_prior\_component\_position\_normal ( *IdxT size*, double *pos\_sigma = default\_sigma\_pos* ) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.6.4.54 **prior\_hessian::TruncatedParetoDist** `mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma )` [static],[inherited]

Definition at line 104 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.6.4.55 **CompositeDist** `mappel::Gauss1DModel::make_prior_normal_position ( IdxT size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg )` [static],[inherited]

Definition at line 114 of file `Gauss1DModel.cpp`.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_normal()`.

Referenced by `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`.

9.6.4.56 **Gauss1DModel::Stencil** `mappel::Gauss1DModel::make_stencil ( const ParamT & theta, bool compute_derivatives = true ) const` [inline],[inherited]

Make a new `Model::Stencil` object at `theta`.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular `theta` (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 116 of file `Gauss1DModel.h`.

References `mappel::Gauss1DModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`.

**9.6.4.57 Gauss1DMAP & mappel::Gauss1DMAP::operator= ( const Gauss1DMAP & o )**

Definition at line 51 of file Gauss1DMAP.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise1DObjective::operator=(), mappel::Gauss1DModel::operator=(), and mappel::PointEmitterModel::operator=().

**9.6.4.58 Gauss1DMAP & mappel::Gauss1DMAP::operator= ( Gauss1DMAP && o )**

Definition at line 62 of file Gauss1DMAP.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise1DObjective::operator=(), mappel::Gauss1DModel::operator=(), and mappel::PointEmitterModel::operator=().

**9.6.4.59 void mappel::Gauss1DModel::pixel\_grad ( IdxT i, const Stencil & s, ParamT & pgrad ) const [inline], [inherited]**

Definition at line 141 of file Gauss1DModel.h.

References mappel::Gauss1DModel::Stencil::DX, mappel::Gauss1DModel::Stencil::l(), and mappel::Gauss1DModel::Stencil::X.

Referenced by mappel::Gauss1DModel::pixel\_hess\_update().

**9.6.4.60 void mappel::Gauss1DModel::pixel\_grad2 ( IdxT i, const Stencil & s, ParamT & pgrad2 ) const [inline], [inherited]**

Definition at line 150 of file Gauss1DModel.h.

References mappel::Gauss1DModel::Stencil::DXS, mappel::Gauss1DModel::Stencil::l(), and mappel::Gauss1DModel::psf\_sigma.

**9.6.4.61 void mappel::Gauss1DModel::pixel\_hess ( IdxT i, const Stencil & s, MatT & hess ) const [inline], [inherited]**

Definition at line 159 of file Gauss1DModel.h.

References mappel::Gauss1DModel::Stencil::DX, mappel::Gauss1DModel::Stencil::DXS, mappel::Gauss1DModel::Stencil::l(), and mappel::Gauss1DModel::psf\_sigma.

**9.6.4.62 void mappel::Gauss1DModel::pixel\_hess\_update ( IdxT i, const Stencil & s, double dm\_ratio\_m1, double dmm\_ratio, ParamT & grad, MatT & hess ) const [inherited]**

pixel derivative inner loop calculations.

Definition at line 191 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::Stencil::DXS, mappel::Gauss1DModel::Stencil::l(), mappel::PointEmitterModel::make\_param(), mappel::Gauss1DModel::pixel\_grad(), and mappel::Gauss1DModel::psf\_sigma.

**9.6.4.63** `double mappel::Gauss1DModel::pixel_model_value ( IdxT i, const Stencil & s ) const` `[inline], [inherited]`

Definition at line 135 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::bg()`, `mappel::Gauss1DModel::Stencil::l()`, and `mappel::Gauss1DModel::Stencil::X`.

**9.6.4.64** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.6.4.65** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.6.4.66** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.6.4.67** `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 108 of file MCMCAdaptor1D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.6.4.68** `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 122 of file MCMCAdaptor1D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.6.4.69 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
[*inherited*]

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.6.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` [*inline*], [*inherited*]

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.6.4.71 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` [*inherited*]

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.6.4.72 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
[*inherited*]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.6.4.73 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` [*inline*],  
[*inherited*]

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.6.4.74 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` [*inline*],  
[*inherited*]

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.6.4.75** `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams ) [inline],[inherited]`

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

**9.6.4.76** `template<class ImT > void ImageFormat1DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]`

Definition at line 115 of file ImageFormat1DBase.h.

**9.6.4.77** `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 ) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

**9.6.4.78** `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

**9.6.4.79** `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases ) [protected],[inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.6.4.80** `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale ) [inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.6.4.81 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc ) [inline], [inherited]`

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.6.4.82 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ ) [inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::set\_prior(), and mappel::Gauss2DsModel::set\_prior().

9.6.4.83 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ ) [inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

9.6.4.84 `void mappel::Gauss1DModel::set_psf_sigma ( double new_psf_sigma ) [inherited]`

Definition at line 125 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::check\_psf\_sigma(), and mappel::Gauss1DModel::psf\_sigma.

Referenced by mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

9.6.4.85 `void mappel::Gauss1DModel::set_psf_sigma ( const VecT & new_psf_sigma ) [inline], [inherited]`

Definition at line 131 of file Gauss1DModel.h.

References mappel::Gauss1DModel::set\_psf\_sigma().

9.6.4.86 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static], [inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.6.4.87 `void ImageFormat1DBase::set_size ( const ImageSizeT & size_ ) [inherited]`

Definition at line 30 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::check\_size(), and mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::set\_size(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DsModel::set\_size().

**9.6.4.88** `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz ) [inline],[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

**9.6.4.89** `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

**9.6.4.90** `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.6.4.91** `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const [inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

## 9.6.5 Member Data Documentation

**9.6.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static],[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

**9.6.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2 [static],[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.



**9.6.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

**9.6.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.6.5.5** `const double mappel::PointEmitterModel::default_max_I = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.6.5.6** `const double mappel::PointEmitterModel::default_mean_I = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.6.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.6.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.6.5.9** `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal" [static],[inherited]`

Definition at line 53 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::operator=()`.

**9.6.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.6.5.11** `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names [static],[inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

**9.6.5.12** `double mappel::MCMCAdaptor1D::eta_bg =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.6.5.13** `double mappel::MCMCAdaptor1D::eta_l =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.6.5.14** `double mappel::MCMCAdaptor1D::eta_x =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

**9.6.5.15** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.6.5.16** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.6.5.17** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.6.5.18** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.6.5.19** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.6.5.20** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

#### 9.6.5.21 **ParamT** mappel::PointEmitterModel::lbound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.6.5.22 **const std::string** mappel::Gauss1DMP::name [static]

Definition at line 34 of file Gauss1DMP.h.

#### 9.6.5.23 **const ImageFormat1DBase::ImageCoordT** ImageFormat1DBase::num\_dim = 1 [static],[inherited]

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::get\_stats().

#### 9.6.5.24 **IdxT** mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.6.5.25 **IdxT** mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.6.5.26 **IdxT** mappel::MCMCAdaptorBase::num\_phases [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().

#### 9.6.5.27 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

#### 9.6.5.28 const StringVecT mappel::Gauss1DModel::prior\_types [static],[inherited]

**Initial value:**

```
= { "Beta",  
                                     "Normal"  
}
```

Definition at line 52 of file Gauss1DModel.h.

Referenced by mappel::Gauss1DModel::operator=().

#### 9.6.5.29 double mappel::Gauss1DModel::psf\_sigma [protected],[inherited]

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 90 of file Gauss1DModel.h.

Referenced by mappel::Gauss1DModel::Stencil::compute\_derivatives(), mappel::Gauss1DModel::get\_psf\_sigma(), mappel::Gauss1DModel::operator=(), mappel::Gauss1DModel::pixel\_grad2(), mappel::Gauss1DModel::pixel\_hess(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss1DModel::Stencil::Stencil().

#### 9.6.5.30 double mappel::MCMCAdaptorBase::sigma\_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

### 9.6.5.31 `ImageSizeT mappel::ImageFormat1DBase::size` [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 65 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

### 9.6.5.32 `ParamT mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

The documentation for this class was generated from the following files:

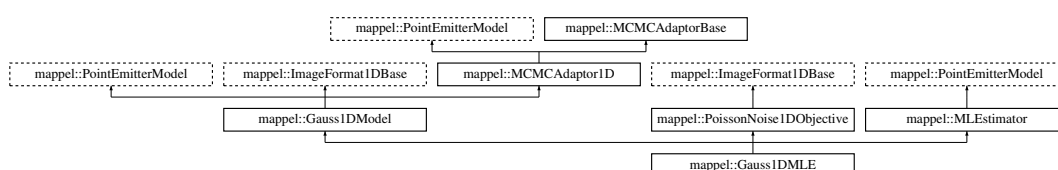
- [Gauss1DMAP.h](#)
- [Gauss1DMAP.cpp](#)

## 9.7 `mappel::Gauss1DMLE` Class Reference

A 1D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DMLE.h>
```

Inheritance diagram for `mappel::Gauss1DMLE`:



## Public Types

- using `StencilVecT` = `std::vector< Stencil >`
- using `ParamT` = `arma::vec`
- using `ParamVecT` = `arma::mat`
- using `ImageCoordT` = `uint32_t`
- using `ImagePixelT` = `double`
- template<class `CoordT` >  
using `ImageSizeShapeT` = `CoordT`
- template<class `CoordT` >  
using `ImageSizeVecShapeT` = `arma::Col< CoordT >`
- using `ImageSizeT` = `ImageSizeShapeT< ImageCoordT >`
- using `ImageSizeVecT` = `ImageSizeVecShapeT< ImageCoordT >`
- template<class `PixelT` >  
using `ImageShapeT` = `arma::Col< PixelT >`
- template<class `PixelT` >  
using `ImageStackShapeT` = `arma::Mat< PixelT >`
- using `ImageT` = `ImageShapeT< ImagePixelT >`
- using `ImageStackT` = `ImageStackShapeT< ImagePixelT >`
- using `ModelDataT` = `ImageT`
- using `ModelDataStackT` = `ImageStackT`

## Public Member Functions

- `Gauss1DMLE` (`arma::Col< ImageCoordT > size`, `VecT psf_sigma`, `const std::string &prior_type=DefaultPriorType`)
- `Gauss1DMLE` (`ImageSizeT size`, `double psf_sigma`, `const std::string &prior_type=DefaultPriorType`)
- `Gauss1DMLE` (`ImageSizeT size`, `double psf_sigma`, `CompositeDist &&prior`)
- `Gauss1DMLE` (`ImageSizeT size`, `double psf_sigma`, `const CompositeDist &prior`)
- `Gauss1DMLE` (`const Gauss1DMLE &o`)
- `Gauss1DMLE & operator=` (`const Gauss1DMLE &o`)
- `Gauss1DMLE` (`Gauss1DMLE &&o`)
- `Gauss1DMLE & operator=` (`Gauss1DMLE &&o`)
- `double get_psf_sigma` () `const`
- `double get_psf_sigma` (`IdxT idx`) `const`
- `void set_psf_sigma` (`double new_psf_sigma`)
- `void set_psf_sigma` (`const VecT &new_psf_sigma`)
- `StatsT get_stats` () `const`
- `Stencil make_stencil` (`const ParamT &theta`, `bool compute_derivatives=true`) `const`  
*Make a new Model::Stencil object at theta.*
- `double pixel_model_value` (`IdxT i`, `const Stencil &s`) `const`
- `void pixel_grad` (`IdxT i`, `const Stencil &s`, `ParamT &pgrad`) `const`
- `void pixel_grad2` (`IdxT i`, `const Stencil &s`, `ParamT &pgrad2`) `const`
- `void pixel_hess` (`IdxT i`, `const Stencil &s`, `MatT &hess`) `const`
- `void pixel_hess_update` (`IdxT i`, `const Stencil &s`, `double dm_ratio_m1`, `double dmm_ratio`, `ParamT &grad`, `MatT &hess`) `const`  
*pixel derivative inner loop calculations.*
- `Stencil initial_theta_estimate` (`const ImageT &im`) `const`  
*Fast, heuristic estimate of initial theta.*
- `Stencil initial_theta_estimate` (`const ImageT &im`, `const ParamT &theta_init`) `const`

- `IdxT get_num_params () const`
- `void check_param_shape (const ParamT &theta) const`
- `void check_param_shape (const ParamVecT &theta) const`
- `void check_psf_sigma (double psf_sigma) const`
- `void check_psf_sigma (const VecT &psf_sigma) const`
- `ParamT make_param () const`
- `template<class FillT >`  
`ParamT make_param (FillT fill) const`
- `ParamVecT make_param_stack (IdxT n) const`
- `template<class FillT >`  
`ParamVecT make_param_stack (IdxT n, FillT fill) const`
- `MatT make_param_mat () const`
- `template<class FillT >`  
`MatT make_param_mat (FillT fill) const`
- `CubeT make_param_mat_stack (IdxT n) const`
- `template<class FillT >`  
`CubeT make_param_mat_stack (IdxT n, FillT fill) const`
- `CompositeDist & get_prior ()`
- `const CompositeDist & get_prior () const`
- `void set_prior (CompositeDist &&prior_)`
- `void set_prior (const CompositeDist &prior_)`
- `IdxT get_num_hyperparams () const`
- `void set_hyperparams (const VecT &hyperparams)`
- `VecT get_hyperparams () const`
- `bool has_hyperparam (const std::string &name) const`
- `double get_hyperparam_value (const std::string &name) const`
- `int get_hyperparam_index (const std::string &name) const`
- `void set_hyperparam_value (const std::string &name, double value)`
- `void rename_hyperparam (const std::string &old_name, const std::string &new_name)`
- `StringVecT get_param_names () const`
- `void set_param_names (const StringVecT &desc)`
- `StringVecT get_hyperparam_names () const`
- `void set_hyperparam_names (const StringVecT &desc)`
- `template<class RngT >`  
`ParamT sample_prior (RngT &rng) const`
- `ParamT sample_prior () const`
- `void set_bounds (const ParamT &lbound, const ParamT &ubound)`
- `void set_lbound (const ParamT &lbound)`
- `void set_ubound (const ParamT &ubound)`
- `const ParamT & get_lbound () const`
- `const ParamT & get_ubound () const`
- `bool theta_in_bounds (const ParamT &theta) const`
- `void bound_theta (ParamT &theta, double epsilon=bounds_epsilon) const`
- `ParamT bounded_theta (const ParamT &theta, double epsilon=bounds_epsilon) const`
- `ParamT reflected_theta (const ParamT &theta) const`
- `BoolVecT theta_stack_in_bounds (const ParamVecT &theta) const`
- `ParamVecT bounded_theta_stack (const ParamVecT &theta, double epsilon=bounds_epsilon) const`
- `ParamVecT reflected_theta_stack (const ParamVecT &theta) const`
- `ImageT make_image () const`
- `ImageStackT make_image_stack (ImageCoordT n) const`
- `ImageCoordT get_size_image_stack (const ImageStackT &stack) const`



- `ImageT get_image_from_stack` (const `ImageStackT` &stack, `ImageCoordT` n) const
- `template<class ImT >`  
`void set_image_in_stack` (`ImageStackT` &stack, `ImageCoordT` n, const `ImT` &im) const
- `ImageSizeT get_size` () const
- `ImageCoordT get_size` (`IdxT` idx) const
- `ImageCoordT get_num_pixels` () const
- `void set_size` (const `ImageSizeT` &size\_)
- `void set_size` (const `arma::Col`< `ImageCoordT` > &sz)
- `void check_image_shape` (const `ImageT` &im) const  
*Check the shape of a single images is correct for model size.*
- `void check_image_shape` (const `ImageStackT` &ims) const  
*Check the shape of a stack of images is correct for model size.*
- `void sample_mcmc_candidate` (`IdxT` sample\_index, `ParamT` &candidate, double step\_scale=1.0) const
- `void sample_mcmc_candidate` (`IdxT` sample\_index, `ParamT` &candidate, const `IdxVecT` &fixed\_parameters\_↔ mask, double step\_scale=1.0) const
- `void set_intensity_mcmc_sampling` (double eta\_l=-1)
- `void set_background_mcmc_sampling` (double eta\_bg=-1)
- `void set_mcmc_sigma_scale` (double scale)
- `double get_mcmc_sigma_scale` () const
- `IdxT get_mcmc_num_phases` () const

#### Static Public Member Functions

- static `CompositeDist make_default_prior` (`IdxT` size, const `std::string` &prior\_type)
- static `CompositeDist make_default_prior_beta_position` (`IdxT` size)
- static `CompositeDist make_default_prior_normal_position` (`IdxT` size)
- static `CompositeDist make_prior_beta_position` (`IdxT` size, double beta\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static `CompositeDist make_prior_normal_position` (`IdxT` size, double sigma\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static `prior_hessian::TruncatedNormalDist make_prior_component_position_normal` (`IdxT` size, double pos\_↔ sigma=default\_sigma\_pos)
- static `prior_hessian::ScaledSymmetricBetaDist make_prior_component_position_beta` (`IdxT` size, double pos\_↔ beta=default\_beta\_pos)
- static `prior_hessian::TruncatedGammaDist make_prior_component_intensity` (double mean=default\_mean\_↔ l, double kappa=default\_intensity\_kappa)
- static `prior_hessian::TruncatedParetoDist make_prior_component_sigma` (double min\_sigma, double max\_↔ sigma, double alpha=default\_alpha\_sigma)
- static `void set_rng_seed` (`RngSeedT` seed)
- static `ParallelRngManagerT & get_rng_manager` ()
- static `ParallelRngGeneratorT & get_rng_generator` ()
- static `void check_size` (const `ImageSizeT` &size\_)

*Check the size argument for the model.*

### Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) = 1
- static const [ImageCoordT](#) [global\\_min\\_size](#) = 3
- static const [ImageCoordT](#) [global\\_max\\_size](#) = 512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5
- static const std::vector< std::string > [estimator\\_names](#)

### Protected Member Functions

- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) [num\\_phases](#))

### Protected Attributes

- double [psf\\_sigma](#)
- CompositeDist [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta\\_x](#) =0
- double [eta\\_l](#) =0
- double [eta\\_bg](#) =0
- [IdxT](#) [num\\_phases](#)
- double [sigma\\_scale](#)

### 9.7.1 Detailed Description

A 1D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective.

Model: [Gauss1DModel](#) - 1D Gaussian PSF with fixed PSF sigma Objective: [PoissonNoise1DObjective](#) - Poisson noise model for 1D Estimator: [MLEstimator](#) - Pure-likelihood estimator

Definition at line 23 of file Gauss1DMLE.h.

### 9.7.2 Member Typedef Documentation

#### 9.7.2.1 using mappel::ImageFormat1DBase::ImageCoordT = uint32\_t [inherited]

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

#### 9.7.2.2 using mappel::ImageFormat1DBase::ImagePixelT = double [inherited]

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

#### 9.7.2.3 template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT> [inherited]

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

#### 9.7.2.4 template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT [inherited]

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

#### 9.7.2.5 using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT> [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

#### 9.7.2.6 template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT> [inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

#### 9.7.2.7 using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT> [inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

**9.7.2.8** `template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

**9.7.2.9** `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

**9.7.2.10** `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

**9.7.2.11** `using mappel::PoissonNoise1DObjective::ModelDataStackT = ImageStackT` `[inherited]`

Objective function data stack type: 1D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 26 of file PoissonNoise1DObjective.h.

**9.7.2.12** `using mappel::PoissonNoise1DObjective::ModelDataT = ImageT` `[inherited]`

Objective function data type: 1D double precision image, gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise1DObjective.h.

**9.7.2.13** `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

**9.7.2.14** `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

**9.7.2.15** `using mappel::Gauss1DModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 49 of file Gauss1DModel.h.

### 9.7.3 Constructor & Destructor Documentation

**9.7.3.1** `mappel::Gauss1DMLE::Gauss1DMLE ( arma::Col< ImageCoordT > size, VecT psf_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 11 of file Gauss1DMLE.cpp.

**9.7.3.2** `mappel::Gauss1DMLE::Gauss1DMLE ( ImageSizeT size, double psf_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 15 of file Gauss1DMLE.cpp.

**9.7.3.3** `mappel::Gauss1DMLE::Gauss1DMLE ( ImageSizeT size, double psf_sigma, CompositeDist && prior )`

Definition at line 19 of file Gauss1DMLE.cpp.

**9.7.3.4** `mappel::Gauss1DMLE::Gauss1DMLE ( ImageSizeT size, double psf_sigma, const CompositeDist & prior )`

Definition at line 27 of file Gauss1DMLE.cpp.

**9.7.3.5** `mappel::Gauss1DMLE::Gauss1DMLE ( const Gauss1DMLE & o )`

Definition at line 35 of file Gauss1DMLE.cpp.

**9.7.3.6** `mappel::Gauss1DMLE::Gauss1DMLE ( Gauss1DMLE && o )`

Definition at line 43 of file Gauss1DMLE.cpp.

### 9.7.4 Member Function Documentation

**9.7.4.1** `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const`  
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

**9.7.4.2** `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

**9.7.4.3** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` `[inherited]`

Definition at line 306 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

**9.7.4.4** `void ImageFormat1DBase::check_image_shape ( const ImageT & im ) const` `[inherited]`

Check the shape of a single images is correct for model size.

Definition at line 59 of file `ImageFormat1DBase.cpp`.

References `mappel::ImageFormat1DBase::size`.

**9.7.4.5** `void ImageFormat1DBase::check_image_shape ( const ImageStackT & ims ) const` `[inherited]`

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file `ImageFormat1DBase.cpp`.

References `mappel::ImageFormat1DBase::size`.

**9.7.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` `[inherited]`

Definition at line 167 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.7.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 176 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::num_params`.

**9.7.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` `[inherited]`

Definition at line 185 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

9.7.4.9 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

9.7.4.10 `void ImageFormat1DBase::check_size ( const ImageSizeT & size_ )` [static],[inherited]

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::global_max_size`, and `mappel::ImageFormat1DBase::global_min_size`.

Referenced by `mappel::ImageFormat1DBase::ImageFormat1DBase()`, and `mappel::ImageFormat1DBase::set_size()`.

9.7.4.11 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline],[inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.7.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` [inline],[inherited]

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.7.4.13 `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.7.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` [inline],[inherited]

Definition at line 224 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.7.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` [inline],[inherited]

Definition at line 108 of file ImageFormat1DBase.h.

**9.7.4.16** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` [inline],  
[inherited]

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.7.4.17** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` [inherited]

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

**9.7.4.18** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` [inherited]

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

**9.7.4.19** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` [inline],[inherited]

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

**9.7.4.20** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` [inline],[inherited]

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.7.4.21** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_num_pixels ( ) const` [inline],  
[inherited]

Definition at line 82 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.7.4.22** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const` [inline],[inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.



**9.7.4.23 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.7.4.24 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.7.4.25 double mappel::Gauss1DModel::get\_psf\_sigma ( ) const** [inline],[inherited]

Definition at line 127 of file Gauss1DModel.h.

References mappel::Gauss1DModel::psf\_sigma.

Referenced by mappel::Gauss1DModel::get\_stats().

**9.7.4.26 double mappel::Gauss1DModel::get\_psf\_sigma ( IdxT idx ) const** [inherited]

Definition at line 131 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::psf\_sigma.

**9.7.4.27 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.7.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( )** [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.7.4.29 ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get\_size ( ) const** [inline],[inherited]

Definition at line 71 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.7.4.30** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size ( IdxT idx ) const` `[inherited]`

Definition at line 20 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::size`.

**9.7.4.31** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack ( const ImageStackT & stack ) const` `[inline],[inherited]`

Definition at line 101 of file ImageFormat1DBase.h.

**9.7.4.32** `StatsT mappel::Gauss1DModel::get_stats ( ) const` `[inherited]`

Definition at line 178 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::get_psf_sigma()`, `mappel::MCMCAdaptor1D::get_stats()`, `mappel::ImageFormat1DBase::get_stats()`, and `mappel::PointEmitterModel::get_stats()`.

**9.7.4.33** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound ( ) const` `[inline],[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DsxyModel::get_max_sigma_ratio()`, `mappel::Gauss2DsModel::get_max_sigma_ratio()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

**9.7.4.34** `bool mappel::PointEmitterModel::has_hyperparam ( const std::string & name ) const` `[inline],[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.7.4.35** `Gauss1DModel::Stencil mappel::Gauss1DModel::initial_theta_estimate ( const ImageT & im ) const` `[inline],[inherited]`

Fast, heuristic estimate of initial theta.

Definition at line 169 of file Gauss1DModel.h.

References `mappel::PointEmitterModel::make_param()`, and `mappel::Gauss1DModel::Stencil::theta`.

**9.7.4.36** `Gauss1DModel::Stencil mappel::Gauss1DModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init ) const` `[inherited]`

Definition at line 207 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::Stencil::bg()`, `mappel::Gauss1DModel::Stencil::l()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::PointEmitterModel::num_params`, and `mappel::ImageFormat1DBase::size`.

**9.7.4.37 CompositeDist** mappel::Gauss1DModel::make\_default\_prior ( *IdxT size*, const std::string & *prior\_type* ) [static],  
[inherited]

Definition at line 59 of file Gauss1DModel.cpp.

References mappel::istarts\_with(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), and mappel::Gauss1DModel::make\_default\_prior\_normal\_position().

**9.7.4.38 CompositeDist** mappel::Gauss1DModel::make\_default\_prior\_beta\_position ( *IdxT size* ) [static],  
[inherited]

Definition at line 80 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component←\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_beta().

Referenced by mappel::Gauss1DModel::make\_default\_prior().

**9.7.4.39 CompositeDist** mappel::Gauss1DModel::make\_default\_prior\_normal\_position ( *IdxT size* ) [static],  
[inherited]

Definition at line 90 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component←\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_normal().

Referenced by mappel::Gauss1DModel::make\_default\_prior().

**9.7.4.40 ImageFormat1DBase::ImageT** ImageFormat1DBase::make\_image ( ) const [inline], [inherited]

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.7.4.41 ImageFormat1DBase::ImageStackT** ImageFormat1DBase::make\_image\_stack ( *ImageCoordT n* ) const  
[inline], [inherited]

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.7.4.42 PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline], [inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess←update().

**9.7.4.43** `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const`  
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.7.4.44** `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.7.4.45** `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.7.4.46** `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.7.4.47** `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`  
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.7.4.48** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

**9.7.4.49** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.7.4.50 **CompositeDist** mappel::Gauss1DModel::make\_prior\_beta\_position ( *IdxT size*, double *beta\_xpos*, double *mean\_l*, double *kappa\_l*, double *mean\_bg*, double *kappa\_bg* ) [static],[inherited]

Definition at line 101 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_beta().

Referenced by mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator().

9.7.4.51 **prior\_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make\_prior\_component\_intensity ( double *mean = default\_mean\_l*, double *kappa = default\_intensity\_kappa* ) [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_l.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.7.4.52 **prior\_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make\_prior\_component\_position\_beta ( *IdxT size*, double *pos\_beta = default\_beta\_pos* ) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), and mappel::Gauss2DsModel::make\_prior\_beta\_position().

9.7.4.53 **prior\_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make\_prior\_component\_position\_normal ( *IdxT size*, double *pos\_sigma = default\_sigma\_pos* ) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.7.4.54 **prior\_hessian::TruncatedParetoDist** `mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma )` [static],[inherited]

Definition at line 104 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.7.4.55 **CompositeDist** `mappel::Gauss1DModel::make_prior_normal_position ( IdxT size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg )` [static],[inherited]

Definition at line 114 of file `Gauss1DModel.cpp`.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_normal()`.

Referenced by `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`.

9.7.4.56 **Gauss1DModel::Stencil** `mappel::Gauss1DModel::make_stencil ( const ParamT & theta, bool compute_derivatives = true ) const` [inline],[inherited]

Make a new `Model::Stencil` object at `theta`.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular `theta` (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 116 of file `Gauss1DModel.h`.

References `mappel::Gauss1DModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`.

**9.7.4.57 Gauss1DMLE & mappel::Gauss1DMLE::operator= ( const Gauss1DMLE & o )**

Definition at line 51 of file Gauss1DMLE.cpp.

References mappel::MLEstimator::operator=(), mappel::PoissonNoise1DObjective::operator=(), mappel::Gauss1DModel::operator=(), and mappel::PointEmitterModel::operator=().

**9.7.4.58 Gauss1DMLE & mappel::Gauss1DMLE::operator= ( Gauss1DMLE && o )**

Definition at line 62 of file Gauss1DMLE.cpp.

References mappel::MLEstimator::operator=(), mappel::PoissonNoise1DObjective::operator=(), mappel::Gauss1DModel::operator=(), and mappel::PointEmitterModel::operator=().

**9.7.4.59 void mappel::Gauss1DModel::pixel\_grad ( IdxT i, const Stencil & s, ParamT & pgrad ) const [inline], [inherited]**

Definition at line 141 of file Gauss1DModel.h.

References mappel::Gauss1DModel::Stencil::DX, mappel::Gauss1DModel::Stencil::l(), and mappel::Gauss1DModel::Stencil::X.

Referenced by mappel::Gauss1DModel::pixel\_hess\_update().

**9.7.4.60 void mappel::Gauss1DModel::pixel\_grad2 ( IdxT i, const Stencil & s, ParamT & pgrad2 ) const [inline], [inherited]**

Definition at line 150 of file Gauss1DModel.h.

References mappel::Gauss1DModel::Stencil::DXS, mappel::Gauss1DModel::Stencil::l(), and mappel::Gauss1DModel::psf\_sigma.

**9.7.4.61 void mappel::Gauss1DModel::pixel\_hess ( IdxT i, const Stencil & s, MatT & hess ) const [inline], [inherited]**

Definition at line 159 of file Gauss1DModel.h.

References mappel::Gauss1DModel::Stencil::DX, mappel::Gauss1DModel::Stencil::DXS, mappel::Gauss1DModel::Stencil::l(), and mappel::Gauss1DModel::psf\_sigma.

**9.7.4.62 void mappel::Gauss1DModel::pixel\_hess\_update ( IdxT i, const Stencil & s, double dm\_ratio\_m1, double dmm\_ratio, ParamT & grad, MatT & hess ) const [inherited]**

pixel derivative inner loop calculations.

Definition at line 191 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::Stencil::DXS, mappel::Gauss1DModel::Stencil::l(), mappel::PointEmitterModel::make\_param(), mappel::Gauss1DModel::pixel\_grad(), and mappel::Gauss1DModel::psf\_sigma.

**9.7.4.63** `double mappel::Gauss1DModel::pixel_model_value ( IdxT i, const Stencil & s ) const` `[inline], [inherited]`

Definition at line 135 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::bg()`, `mappel::Gauss1DModel::Stencil::l()`, and `mappel::Gauss1DModel::Stencil::X`.

**9.7.4.64** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.7.4.65** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.7.4.66** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.7.4.67** `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 108 of file MCMCAdaptor1D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.7.4.68** `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 122 of file MCMCAdaptor1D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.



9.7.4.69 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
[*inherited*]

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.7.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` [*inline*], [*inherited*]

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.7.4.71 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` [*inherited*]

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.7.4.72 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
[*inherited*]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.7.4.73 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` [*inline*],  
[*inherited*]

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.7.4.74 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` [*inline*],  
[*inherited*]

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.7.4.75** `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams ) [inline],[inherited]`

Definition at line 220 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::set\_hyperparams(), and mappel::Gauss2DsModel::set\_hyperparams().

**9.7.4.76** `template<class ImT > void ImageFormat1DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]`

Definition at line 115 of file ImageFormat1DBase.h.

**9.7.4.77** `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 ) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_mean\_I, mappel::MCMCAdaptor1D::eta\_I, mappel::PointEmitterModel::get\_hyperparam\_value(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

**9.7.4.78** `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

**9.7.4.79** `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases ) [protected],[inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

Referenced by mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), and mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds().

**9.7.4.80** `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale ) [inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::global\_max\_mcmc\_sigma\_scale, and mappel::MCMCAdaptorBase::sigma\_scale.

9.7.4.81 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc ) [inline], [inherited]`

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.7.4.82 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ ) [inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::set\_prior(), and mappel::Gauss2DsModel::set\_prior().

9.7.4.83 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ ) [inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

9.7.4.84 `void mappel::Gauss1DModel::set_psf_sigma ( double new_psf_sigma ) [inherited]`

Definition at line 125 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::check\_psf\_sigma(), and mappel::Gauss1DModel::psf\_sigma.

Referenced by mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

9.7.4.85 `void mappel::Gauss1DModel::set_psf_sigma ( const VecT & new_psf_sigma ) [inline], [inherited]`

Definition at line 131 of file Gauss1DModel.h.

References mappel::Gauss1DModel::set\_psf\_sigma().

9.7.4.86 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static], [inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.7.4.87 `void ImageFormat1DBase::set_size ( const ImageSizeT & size_ ) [inherited]`

Definition at line 30 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::check\_size(), and mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::set\_size(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DsModel::set\_size().

**9.7.4.88** `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz ) [inline], [inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

**9.7.4.89** `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

**9.7.4.90** `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.7.4.91** `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const [inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

## 9.7.5 Member Data Documentation

**9.7.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

**9.7.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2 [static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

**9.7.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

**9.7.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.7.5.5** `const double mappel::PointEmitterModel::default_max_I = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.7.5.6** `const double mappel::PointEmitterModel::default_mean_I = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.7.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.7.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.7.5.9** `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal" [static],[inherited]`

Definition at line 53 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::operator=()`.

**9.7.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.7.5.11** `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names [static],[inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

**9.7.5.12** `double mappel::MCMCAdaptor1D::eta_bg =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.7.5.13** `double mappel::MCMCAdaptor1D::eta_l =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.7.5.14** `double mappel::MCMCAdaptor1D::eta_x =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

**9.7.5.15** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.7.5.16** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.7.5.17** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.7.5.18** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.7.5.19** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.7.5.20** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

#### 9.7.5.21 **ParamT** mappel::PointEmitterModel::lbound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.7.5.22 **const std::string** mappel::Gauss1DMLE::name [static]

Definition at line 34 of file Gauss1DMLE.h.

#### 9.7.5.23 **const ImageFormat1DBase::ImageCoordT** ImageFormat1DBase::num\_dim = 1 [static],[inherited]

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::get\_stats().

#### 9.7.5.24 **IdxT** mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.7.5.25 **IdxT** mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.7.5.26 **IdxT** mappel::MCMCAdaptorBase::num\_phases [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().



#### 9.7.5.27 CompositeDist mappel::PointEmitterModel::prior [protected], [inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

#### 9.7.5.28 const StringVecT mappel::Gauss1DModel::prior\_types [static], [inherited]

**Initial value:**

```
= { "Beta",  
                                     "Normal"  
}
```

Definition at line 52 of file Gauss1DModel.h.

Referenced by mappel::Gauss1DModel::operator=().

#### 9.7.5.29 double mappel::Gauss1DModel::psf\_sigma [protected], [inherited]

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 90 of file Gauss1DModel.h.

Referenced by mappel::Gauss1DModel::Stencil::compute\_derivatives(), mappel::Gauss1DModel::get\_psf\_sigma(), mappel::Gauss1DModel::operator=(), mappel::Gauss1DModel::pixel\_grad2(), mappel::Gauss1DModel::pixel\_hess(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss1DModel::Stencil::Stencil().

#### 9.7.5.30 double mappel::MCMCAdaptorBase::sigma\_scale [protected], [inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

### 9.7.5.31 `ImageSizeT mappel::ImageFormat1DBase::size` [protected], [inherited]

Number of pixels in X dimension for 1D image

Definition at line 65 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

### 9.7.5.32 `ParamT mappel::PointEmitterModel::ubound` [protected], [inherited]

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

The documentation for this class was generated from the following files:

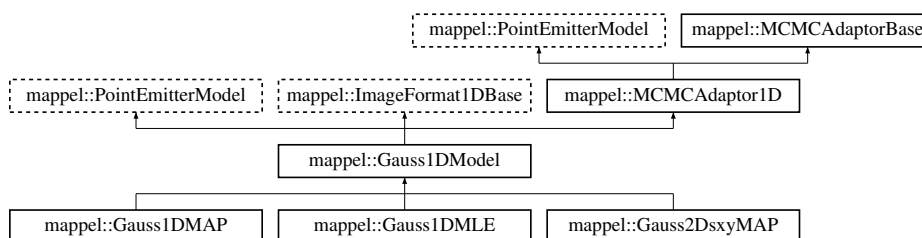
- [Gauss1DMLE.h](#)
- [Gauss1DMLE.cpp](#)

## 9.8 `mappel::Gauss1DModel` Class Reference

A base class for 1D Gaussian PSF with a fixed sigma (standard dev.)

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DModel.h>
```

Inheritance diagram for `mappel::Gauss1DModel`:



### Classes

- class [Stencil](#)  
*Stencil for 1D fixed-sigma models.*

## Public Types

- using `StencilVecT` = `std::vector< Stencil >`
- using `ParamT` = `arma::vec`
- using `ParamVecT` = `arma::mat`
- using `ImageCoordT` = `uint32_t`
- using `ImagePixelT` = `double`
- template<class CoordT >  
using `ImageSizeShapeT` = `CoordT`
- template<class CoordT >  
using `ImageSizeVecShapeT` = `arma::Col< CoordT >`
- using `ImageSizeT` = `ImageSizeShapeT< ImageCoordT >`
- using `ImageSizeVecT` = `ImageSizeVecShapeT< ImageCoordT >`
- template<class PixelT >  
using `ImageShapeT` = `arma::Col< PixelT >`
- template<class PixelT >  
using `ImageStackShapeT` = `arma::Mat< PixelT >`
- using `ImageT` = `ImageShapeT< ImagePixelT >`
- using `ImageStackT` = `ImageStackShapeT< ImagePixelT >`

## Public Member Functions

- double `get_psf_sigma` () const
- double `get_psf_sigma` (IdxT idx) const
- void `set_psf_sigma` (double new\_psf\_sigma)
- void `set_psf_sigma` (const VecT &new\_psf\_sigma)
- StatsT `get_stats` () const
- Stencil `make_stencil` (const ParamT &theta, bool compute\_derivatives=true) const  
*Make a new Model::Stencil object at theta.*
- double `pixel_model_value` (IdxT i, const Stencil &s) const
- void `pixel_grad` (IdxT i, const Stencil &s, ParamT &pgrad) const
- void `pixel_grad2` (IdxT i, const Stencil &s, ParamT &pgrad2) const
- void `pixel_hess` (IdxT i, const Stencil &s, MatT &hess) const
- void `pixel_hess_update` (IdxT i, const Stencil &s, double dm\_ratio\_m1, double dmm\_ratio, ParamT &grad, MatT &hess) const  
*pixel derivative inner loop calculations.*
- Stencil `initial_theta_estimate` (const ImageT &im) const  
*Fast, heuristic estimate of initial theta.*
- Stencil `initial_theta_estimate` (const ImageT &im, const ParamT &theta\_init) const
- IdxT `get_num_params` () const
- void `check_param_shape` (const ParamT &theta) const
- void `check_param_shape` (const ParamVecT &theta) const
- void `check_psf_sigma` (double psf\_sigma) const
- void `check_psf_sigma` (const VecT &psf\_sigma) const
- ParamT `make_param` () const
- template<class FillT >  
`ParamT make_param` (FillT fill) const
- ParamVecT `make_param_stack` (IdxT n) const
- template<class FillT >  
`ParamVecT make_param_stack` (IdxT n, FillT fill) const

- [MatT make\\_param\\_mat \(\)](#) const
- [template<class FillT >](#)  
[MatT make\\_param\\_mat \(FillT fill\)](#) const
- [CubeT make\\_param\\_mat\\_stack \(IdxT n\)](#) const
- [template<class FillT >](#)  
[CubeT make\\_param\\_mat\\_stack \(IdxT n, FillT fill\)](#) const
- [CompositeDist & get\\_prior \(\)](#)
- [const CompositeDist & get\\_prior \(\)](#) const
- [void set\\_prior \(CompositeDist &&prior\\_\)](#)
- [void set\\_prior \(const CompositeDist &prior\\_\)](#)
- [IdxT get\\_num\\_hyperparams \(\)](#) const
- [void set\\_hyperparams \(const Vect &hyperparams\)](#)
- [Vect get\\_hyperparams \(\)](#) const
- [bool has\\_hyperparam \(const std::string &name\)](#) const
- [double get\\_hyperparam\\_value \(const std::string &name\)](#) const
- [int get\\_hyperparam\\_index \(const std::string &name\)](#) const
- [void set\\_hyperparam\\_value \(const std::string &name, double value\)](#)
- [void rename\\_hyperparam \(const std::string &old\\_name, const std::string &new\\_name\)](#)
- [StringVecT get\\_param\\_names \(\)](#) const
- [void set\\_param\\_names \(const StringVecT &desc\)](#)
- [StringVecT get\\_hyperparam\\_names \(\)](#) const
- [void set\\_hyperparam\\_names \(const StringVecT &desc\)](#)
- [template<class RngT >](#)  
[ParamT sample\\_prior \(RngT &rng\)](#) const
- [ParamT sample\\_prior \(\)](#) const
- [void set\\_bounds \(const ParamT &lbound, const ParamT &ubound\)](#)
- [void set\\_lbound \(const ParamT &lbound\)](#)
- [void set\\_ubound \(const ParamT &ubound\)](#)
- [const ParamT & get\\_lbound \(\)](#) const
- [const ParamT & get\\_ubound \(\)](#) const
- [bool theta\\_in\\_bounds \(const ParamT &theta\)](#) const
- [void bound\\_theta \(ParamT &theta, double epsilon=bounds\\_epsilon\)](#) const
- [ParamT bounded\\_theta \(const ParamT &theta, double epsilon=bounds\\_epsilon\)](#) const
- [ParamT reflected\\_theta \(const ParamT &theta\)](#) const
- [BoolVecT theta\\_stack\\_in\\_bounds \(const ParamVecT &theta\)](#) const
- [ParamVecT bounded\\_theta\\_stack \(const ParamVecT &theta, double epsilon=bounds\\_epsilon\)](#) const
- [ParamVecT reflected\\_theta\\_stack \(const ParamVecT &theta\)](#) const
- [ImageT make\\_image \(\)](#) const
- [ImageStackT make\\_image\\_stack \(ImageCoordT n\)](#) const
- [ImageCoordT get\\_size\\_image\\_stack \(const ImageStackT &stack\)](#) const
- [ImageT get\\_image\\_from\\_stack \(const ImageStackT &stack, ImageCoordT n\)](#) const
- [template<class ImT >](#)  
[void set\\_image\\_in\\_stack \(ImageStackT &stack, ImageCoordT n, const ImT &im\)](#) const
- [ImageSizeT get\\_size \(\)](#) const
- [ImageCoordT get\\_size \(IdxT idx\)](#) const
- [ImageCoordT get\\_num\\_pixels \(\)](#) const
- [void set\\_size \(const ImageSizeT &size\\_\)](#)
- [void set\\_size \(const arma::Col< ImageCoordT > &sz\)](#)
- [void check\\_image\\_shape \(const ImageT &im\)](#) const  
*Check the shape of a single images is correct for model size.*
- [void check\\_image\\_shape \(const ImageStackT &ims\)](#) const

*Check the shape of a stack of images is correct for model size.*

- void [sample\\_mcmc\\_candidate](#) (IdxT sample\_index, ParamT &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) (IdxT sample\_index, ParamT &candidate, const IdxVecT &fixed\_parameters\_↵ mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)
- void [set\\_background\\_mcmc\\_sampling](#) (double eta\_bg=-1)
- void [set\\_mcmc\\_sigma\\_scale](#) (double scale)
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- IdxT [get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Member Functions

- static CompositeDist [make\\_default\\_prior](#) (IdxT size, const std::string &prior\_type)
- static CompositeDist [make\\_default\\_prior\\_beta\\_position](#) (IdxT size)
- static CompositeDist [make\\_default\\_prior\\_normal\\_position](#) (IdxT size)
- static CompositeDist [make\\_prior\\_beta\\_position](#) (IdxT size, double beta\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static CompositeDist [make\\_prior\\_normal\\_position](#) (IdxT size, double sigma\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static prior\_hessian::TruncatedNormalDist [make\\_prior\\_component\\_position\\_normal](#) (IdxT size, double pos\_↵ sigma=default\_sigma\_pos)
- static prior\_hessian::ScaledSymmetricBetaDist [make\\_prior\\_component\\_position\\_beta](#) (IdxT size, double pos\_↵ beta=default\_beta\_pos)
- static prior\_hessian::TruncatedGammaDist [make\\_prior\\_component\\_intensity](#) (double mean=default\_mean\_↵ l, double kappa=default\_intensity\_kappa)
- static prior\_hessian::TruncatedParetoDist [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵ sigma, double alpha=default\_alpha\_sigma)
- static void [set\\_rng\\_seed](#) (RngSeedT seed)
- static ParallelRngManagerT & [get\\_rng\\_manager](#) ()
- static ParallelRngGeneratorT & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const ImageSizeT &size\_)

*Check the size argument for the model.*

#### Static Public Attributes

- static const StringVecT [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const ImageCoordT [num\\_dim](#) = 1
- static const ImageCoordT [global\\_min\\_size](#) = 3
- static const ImageCoordT [global\\_max\\_size](#) = 512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5

**Protected Member Functions**

- [Gauss1DModel](#) ([IdxT](#) size, double [psf\\_sigma](#))
- [Gauss1DModel](#) (const [Gauss1DModel](#) &o)
- [Gauss1DModel](#) ([Gauss1DModel](#) &&o)
- [Gauss1DModel](#) & operator= (const [Gauss1DModel](#) &o)
- [Gauss1DModel](#) & operator= ([Gauss1DModel](#) &&o)
- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) num\_phases)

**Protected Attributes**

- double [psf\\_sigma](#)
- CompositeDist [prior](#)
- [IdxT](#) num\_params
- [IdxT](#) num\_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double [eta\\_x](#) =0
- double [eta\\_l](#) =0
- double [eta\\_bg](#) =0
- [IdxT](#) num\_phases
- double [sigma\\_scale](#)

**9.8.1 Detailed Description**

A base class for 1D Gaussian PSF with a fixed sigma (standard dev.)

This base class defines the [Stencil](#) type for 1D Gaussian PSF as well as the prior shape and parameters.

Initialized by an integer, size, and double, [psf\\_sigma](#).

Definition at line 24 of file [Gauss1DModel.h](#).

**9.8.2 Member Typedef Documentation****9.8.2.1 using [mappel::ImageFormat1DBase::ImageCoordT](#) = [uint32\\_t](#) [inherited]**

Image size coordinate storage type

Definition at line 25 of file [ImageFormat1DBase.h](#).

**9.8.2.2 using [mappel::ImageFormat1DBase::ImagePixelT](#) = [double](#) [inherited]**

Image pixel storage type

Definition at line 26 of file [ImageFormat1DBase.h](#).

**9.8.2.3** `template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT>`  
[*inherited*]

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

**9.8.2.4** `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT` [*inherited*]

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

**9.8.2.5** `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [*inherited*]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

**9.8.2.6** `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT>`  
[*inherited*]

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

**9.8.2.7** `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
[*inherited*]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

**9.8.2.8** `template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>`  
[*inherited*]

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

**9.8.2.9** `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
[*inherited*]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

**9.8.2.10** `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

**9.8.2.11** `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

**9.8.2.12** `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

**9.8.2.13** `using mappel::Gauss1DModel::StencilVecT = std::vector<Stencil>`

Definition at line 49 of file Gauss1DModel.h.

### 9.8.3 Constructor & Destructor Documentation

**9.8.3.1** `mappel::Gauss1DModel::Gauss1DModel ( IdxT size, double psf_sigma )` [protected]

Definition at line 12 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`.

**9.8.3.2** `mappel::Gauss1DModel::Gauss1DModel ( const Gauss1DModel & o )` [protected]

Definition at line 20 of file Gauss1DModel.cpp.

**9.8.3.3** `mappel::Gauss1DModel::Gauss1DModel ( Gauss1DModel && o )` [protected]

Definition at line 26 of file Gauss1DModel.cpp.

### 9.8.4 Member Function Documentation

**9.8.4.1** `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const`  
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.



**9.8.4.2** `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

**9.8.4.3** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

**9.8.4.4** `void ImageFormat1DBase::check_image_shape ( const ImageT & im ) const` [inherited]

Check the shape of a single images is correct for model size.

Definition at line 59 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::size`.

**9.8.4.5** `void ImageFormat1DBase::check_image_shape ( const ImageStackT & ims ) const` [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::size`.

**9.8.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.8.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

**9.8.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` `[inherited]`

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

**9.8.4.9** `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

**9.8.4.10** `void ImageFormat1DBase::check_size ( const ImageSizeT & size_ )` `[static],[inherited]`

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::global_max_size`, and `mappel::ImageFormat1DBase::global_min_size`.

Referenced by `mappel::ImageFormat1DBase::ImageFormat1DBase()`, and `mappel::ImageFormat1DBase::set_size()`.

**9.8.4.11** `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline],[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.8.4.12** `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` `[inline],[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.8.4.13** `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline],[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.8.4.14** `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const [inline], [inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.8.4.15** `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const [inline], [inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

**9.8.4.16** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const [inline], [inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.8.4.17** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const [inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

**9.8.4.18** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const [inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

**9.8.4.19** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const [inline], [inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

**9.8.4.20** `IdxT mappel::PointEmitterModel::get_num_params ( ) const [inline], [inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.8.4.21 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get\_num\_pixels ( ) const** [inline],  
[inherited]

Definition at line 82 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.8.4.22 StringVecT mappel::PointEmitterModel::get\_param\_names ( ) const** [inline],[inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.8.4.23 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel↵  
::update\_internal\_1Dsum\_estimators().

**9.8.4.24 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.8.4.25 double mappel::Gauss1DModel::get\_psf\_sigma ( ) const** [inline]

Definition at line 127 of file Gauss1DModel.h.

References psf\_sigma.

Referenced by get\_stats().

**9.8.4.26 double mappel::Gauss1DModel::get\_psf\_sigma ( IdxT idx ) const**

Definition at line 131 of file Gauss1DModel.cpp.

References psf\_sigma.

**9.8.4.27 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.8.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( ) [static],[inherited]**

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.8.4.29 ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get\_size ( ) const [inline],[inherited]**

Definition at line 71 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.8.4.30 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get\_size ( IdxT idx ) const [inherited]**

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

**9.8.4.31 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get\_size\_image\_stack ( const ImageStackT & stack ) const [inline],[inherited]**

Definition at line 101 of file ImageFormat1DBase.h.

**9.8.4.32 StatsT mappel::Gauss1DModel::get\_stats ( ) const**

Definition at line 178 of file Gauss1DModel.cpp.

References get\_psf\_sigma(), mappel::MCMCAdaptor1D::get\_stats(), mappel::ImageFormat1DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

**9.8.4.33 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const [inline],[inherited]**

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

**9.8.4.34 bool mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const [inline],[inherited]**

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.8.4.35** `Gauss1DModel::Stencil mappel::Gauss1DModel::initial_theta_estimate ( const ImageT & im ) const` `[inline]`

Fast, heuristic estimate of initial theta.

Definition at line 169 of file Gauss1DModel.h.

References `mappel::PointEmitterModel::make_param()`, and `mappel::Gauss1DModel::Stencil::theta`.

**9.8.4.36** `Gauss1DModel::Stencil mappel::Gauss1DModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init ) const`

Definition at line 207 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::Stencil::bg()`, `mappel::Gauss1DModel::Stencil::l()`, `make_stencil()`, `mappel::PointEmitterModel::num_params`, and `mappel::ImageFormat1DBase::size`.

**9.8.4.37** `CompositeDist mappel::Gauss1DModel::make_default_prior ( IdxT size, const std::string & prior_type )` `[static]`

Definition at line 59 of file Gauss1DModel.cpp.

References `mappel::istarts_with()`, `make_default_prior_beta_position()`, and `make_default_prior_normal_position()`.

**9.8.4.38** `CompositeDist mappel::Gauss1DModel::make_default_prior_beta_position ( IdxT size )` `[static]`

Definition at line 80 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_beta()`.

Referenced by `make_default_prior()`.

**9.8.4.39** `CompositeDist mappel::Gauss1DModel::make_default_prior_normal_position ( IdxT size )` `[static]`

Definition at line 90 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_normal()`.

Referenced by `make_default_prior()`.

**9.8.4.40** `ImageFormat1DBase::ImageT ImageFormat1DBase::make_image ( ) const` `[inline],[inherited]`

Definition at line 87 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

**9.8.4.41** `ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack ( ImageCoordT n ) const` `[inline],[inherited]`

Definition at line 94 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

**9.8.4.42** `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( ) const` `[inline],[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

**9.8.4.43** `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const` `[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.8.4.44** `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.8.4.45** `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.8.4.46** `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.8.4.47** `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.8.4.48** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.8.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` [inherited]

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.8.4.50 `CompositeDist mappel::Gauss1DModel::make_prior_beta_position ( IdxT size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg )` [static]

Definition at line 101 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_beta().

Referenced by mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator().

9.8.4.51 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_l, double kappa = default_intensity_kappa )` [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_l.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.8.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos )` [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), make\_prior\_beta\_position(), and mappel::Gauss2DsModel::make\_prior\_beta\_position().

9.8.4.53 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos )` [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().



9.8.4.54 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma )` [static],[inherited]

Definition at line 104 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.8.4.55 `CompositeDist mappel::Gauss1DModel::make_prior_normal_position ( IdxT size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg )` [static]

Definition at line 114 of file `Gauss1DModel.cpp`.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_normal()`.

Referenced by `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`.

9.8.4.56 `Gauss1DModel::Stencil mappel::Gauss1DModel::make_stencil ( const ParamT & theta, bool compute_derivatives = true ) const` [inline]

Make a new `Model::Stencil` object at `theta`.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular `theta` (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 116 of file `Gauss1DModel.h`.

References `mappel::Gauss1DModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

Referenced by `initial_theta_estimate()`.

**9.8.4.57** `Gauss1DModel & mappel::Gauss1DModel::operator= ( const Gauss1DModel & o )` `[protected]`

Definition at line 32 of file Gauss1DModel.cpp.

References `mappel::MCMCAdaptor1D::operator=()`, and `psf_sigma`.

Referenced by `mappel::Gauss1DModel::operator=()`, and `mappel::Gauss1DModel::operator=()`.

**9.8.4.58** `Gauss1DModel & mappel::Gauss1DModel::operator= ( Gauss1DModel && o )` `[protected]`

Definition at line 41 of file Gauss1DModel.cpp.

References `DefaultPriorType`, `mappel::MCMCAdaptor1D::operator=()`, `prior_types`, and `psf_sigma`.

**9.8.4.59** `void mappel::Gauss1DModel::pixel_grad ( IdxT i, const Stencil & s, ParamT & pgrad ) const` `[inline]`

Definition at line 141 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::DX`, `mappel::Gauss1DModel::Stencil::l()`, and `mappel::Gauss1DModel::Stencil::X`.

Referenced by `pixel_hess_update()`.

**9.8.4.60** `void mappel::Gauss1DModel::pixel_grad2 ( IdxT i, const Stencil & s, ParamT & pgrad2 ) const` `[inline]`

Definition at line 150 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::DXS`, `mappel::Gauss1DModel::Stencil::l()`, and `psf_sigma`.

**9.8.4.61** `void mappel::Gauss1DModel::pixel_hess ( IdxT i, const Stencil & s, MatT & hess ) const` `[inline]`

Definition at line 159 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::DX`, `mappel::Gauss1DModel::Stencil::DXS`, `mappel::Gauss1DModel::Stencil::l()`, and `psf_sigma`.

**9.8.4.62** `void mappel::Gauss1DModel::pixel_hess_update ( IdxT i, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess ) const`

pixel derivative inner loop calculations.

Definition at line 191 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::Stencil::DXS`, `mappel::Gauss1DModel::Stencil::l()`, `mappel::PointEmitterModel::make_param()`, `pixel_grad()`, and `psf_sigma`.

**9.8.4.63** `double mappel::Gauss1DModel::pixel_model_value ( IdxT i, const Stencil & s ) const` `[inline]`

Definition at line 135 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::bg()`, `mappel::Gauss1DModel::Stencil::l()`, and `mappel::Gauss1DModel::Stencil::X`.

**9.8.4.64** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.8.4.65** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const`  
`[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.8.4.66** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.8.4.67** `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const`  
`[inherited]`

Definition at line 108 of file MCMCAdaptor1D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.8.4.68** `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const`  
`[inherited]`

Definition at line 122 of file MCMCAdaptor1D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.8.4.69** `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.8.4.70** `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline],[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

**9.8.4.71** `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` `[inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

**9.8.4.72** `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.8.4.73** `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline],[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.8.4.74** `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline],[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.8.4.75** `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams )` `[inline],[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

9.8.4.76 `template<class ImT > void ImageFormat1DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const` `[inherited]`

Definition at line 115 of file ImageFormat1DBase.h.

9.8.4.77 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 )` `[inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.8.4.78 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.8.4.79 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` `[protected]`, `[inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.8.4.80 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` `[inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.8.4.81 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` `[inline]`, `[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.8.4.82 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )` `[inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

**9.8.4.83** `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ )` `[inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.8.4.84** `void mappel::Gauss1DModel::set_psf_sigma ( double new_psf_sigma )`

Definition at line 125 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, and `psf_sigma`.

Referenced by `set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

**9.8.4.85** `void mappel::Gauss1DModel::set_psf_sigma ( const VecT & new_psf_sigma )` `[inline]`

Definition at line 131 of file Gauss1DModel.h.

References `set_psf_sigma()`.

**9.8.4.86** `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` `[static]`, `[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

**9.8.4.87** `void ImageFormat1DBase::set_size ( const ImageSizeT & size_ )` `[inherited]`

Definition at line 30 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::check_size()`, and `mappel::ImageFormat1DBase::size`.

Referenced by `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss2DModel::set_size()`, and `mappel::Gauss2DsModel::set_size()`.

**9.8.4.88** `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz )` `[inline]`, `[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

**9.8.4.89** `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

9.8.4.90 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const` `[inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxyModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

9.8.4.91 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

## 9.8.5 Member Data Documentation

9.8.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static]`, `[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

9.8.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static]`, `[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.8.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static]`, `[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.8.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static]`, `[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.8.5.5** `const double mappel::PointEmitterModel::default_max_I = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.8.5.6** `const double mappel::PointEmitterModel::default_mean_I = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.8.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.8.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.8.5.9** `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal"` `[static]`

Definition at line 53 of file Gauss1DModel.h.

Referenced by `operator=()`.

**9.8.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.



**9.8.5.11** `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.8.5.12** `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.8.5.13** `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

**9.8.5.14** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static]`, `[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.8.5.15** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static]`, `[inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.8.5.16** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.8.5.17** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.8.5.18** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.8.5.19** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.8.5.20** `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.8.5.21** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static], [inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.8.5.22** `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

**9.8.5.23** `IdxT mappel::PointEmitterModel::num_params` `[protected], [inherited]`

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

**9.8.5.24** `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected], [inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().

**9.8.5.25** `CompositeDist mappel::PointEmitterModel::prior` `[protected], [inherited]`

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

**9.8.5.26** `const StringVecT mappel::Gauss1DModel::prior_types` `[static]`

**Initial value:**

```
= { "Beta",  
                                     "Normal"  
}
```

Definition at line 52 of file Gauss1DModel.h.

Referenced by operator=().

**9.8.5.27** `double mappel::Gauss1DModel::psf_sigma` `[protected]`

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 90 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `get_psf_sigma()`, `operator=()`, `pixel_grad2()`, `pixel_hess()`, `pixel_hess_update()`, `set_psf_sigma()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

**9.8.5.28** `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`, `[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.8.5.29** `ImageSizeT mappel::ImageFormat1DBase::size` `[protected]`, `[inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

## 9.8.5.30 ParamT mappel::PointEmitterModel::ubound [protected], [inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

The documentation for this class was generated from the following files:

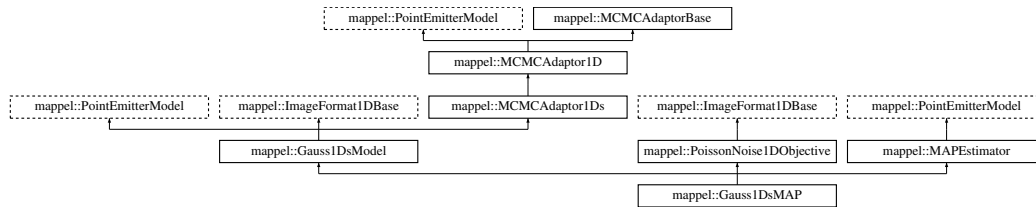
- [Gauss1DModel.h](#)
- [Gauss1DModel.cpp](#)

## 9.9 mappel::Gauss1DsMAP Class Reference

A 1D Gaussian with variable PSF sigma under an Poisson read noise assumption and MAP Objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DsMAP.h>
```

Inheritance diagram for mappel::Gauss1DsMAP:



## Public Types

- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = CoordT
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Col< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Col< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Mat< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >
- using [ModelDataT](#) = [ImageT](#)
- using [ModelDataStackT](#) = [ImageStackT](#)

## Public Member Functions

- [Gauss1DsMAP](#) (arma::Col< [ImageCoordT](#) > [size](#), [VecT](#) [min\\_sigma](#), [VecT](#) [max\\_sigma](#), const std::string &[prior](#)↵  
\_type=[DefaultPriorType](#))
- [Gauss1DsMAP](#) ([ImageSizeT](#) [size](#), double [min\\_sigma](#), double [max\\_sigma](#), const std::string &[prior](#),\_type=[Default](#)↵  
[PriorType](#))
- [Gauss1DsMAP](#) ([ImageSizeT](#) [size](#), [CompositeDist](#) &&[prior](#))
- [Gauss1DsMAP](#) ([ImageSizeT](#) [size](#), const [CompositeDist](#) &[prior](#))
- [Gauss1DsMAP](#) (const [Gauss1DsMAP](#) &o)
- [Gauss1DsMAP](#) & [operator=](#) (const [Gauss1DsMAP](#) &o)
- [Gauss1DsMAP](#) ([Gauss1DsMAP](#) &&o)
- [Gauss1DsMAP](#) & [operator=](#) ([Gauss1DsMAP](#) &&o)
- double [get\\_min\\_sigma](#) () const
- double [get\\_max\\_sigma](#) () const
- void [set\\_min\\_sigma](#) (double [min\\_sigma](#))
- void [set\\_min\\_sigma](#) (const [VecT](#) &[min\\_sigma](#))
- void [set\\_max\\_sigma](#) (double [max\\_sigma](#))
- void [set\\_max\\_sigma](#) (const [VecT](#) &[max\\_sigma](#))
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &[theta](#), bool [compute\\_derivatives](#)=true) const  
*Make a new Model::Stencil object at theta.*
- double [pixel\\_model\\_value](#) ([IdxT](#) i, const [Stencil](#) &s) const
- void [pixel\\_grad](#) ([IdxT](#) i, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel\\_grad2](#) ([IdxT](#) i, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel\\_hess](#) ([IdxT](#) i, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel\\_hess\\_update](#) ([IdxT](#) i, const [Stencil](#) &s, double [dm\\_ratio\\_m1](#), double [dmm\\_ratio](#), [ParamT](#) &[grad](#), [MatT](#) &hess) const
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im) const  
*Fast, heuristic estimate of initial theta.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &[theta\\_init](#)) const
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &[theta](#)) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &[theta](#)) const
- void [check\\_psf\\_sigma](#) (double [psf\\_sigma](#)) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &[psf\\_sigma](#)) const
- [ParamT](#) [make\\_param](#) () const
- template<class [FillT](#) >  
  [ParamT](#) [make\\_param](#) ([FillT](#) fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class [FillT](#) >  
  [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n, [FillT](#) fill) const
- [MatT](#) [make\\_param\\_mat](#) () const
- template<class [FillT](#) >  
  [MatT](#) [make\\_param\\_mat](#) ([FillT](#) fill) const
- [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class [FillT](#) >  
  [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n, [FillT](#) fill) const
- [CompositeDist](#) & [get\\_prior](#) ()
- const [CompositeDist](#) & [get\\_prior](#) () const
- void [set\\_prior](#) ([CompositeDist](#) &&[prior](#)\_)
- void [set\\_prior](#) (const [CompositeDist](#) &[prior](#)\_)

- `IdxT get_num_hyperparams () const`
- `void set_hyperparams (const Vect &hyperparams)`
- `Vect get_hyperparams () const`
- `bool has_hyperparam (const std::string &name) const`
- `double get_hyperparam_value (const std::string &name) const`
- `int get_hyperparam_index (const std::string &name) const`
- `void set_hyperparam_value (const std::string &name, double value)`
- `void rename_hyperparam (const std::string &old_name, const std::string &new_name)`
- `StringVecT get_param_names () const`
- `void set_param_names (const StringVecT &desc)`
- `StringVecT get_hyperparam_names () const`
- `void set_hyperparam_names (const StringVecT &desc)`
- `template<class RngT >`  
`ParamT sample_prior (RngT &rng) const`
- `ParamT sample_prior () const`
- `void set_bounds (const ParamT &lbound, const ParamT &ubound)`
- `void set_lbound (const ParamT &lbound)`
- `void set_ubound (const ParamT &ubound)`
- `const ParamT &get_lbound () const`
- `const ParamT &get_ubound () const`
- `bool theta_in_bounds (const ParamT &theta) const`
- `void bound_theta (ParamT &theta, double epsilon=bounds_epsilon) const`
- `ParamT bounded_theta (const ParamT &theta, double epsilon=bounds_epsilon) const`
- `ParamT reflected_theta (const ParamT &theta) const`
- `BoolVecT theta_stack_in_bounds (const ParamVecT &theta) const`
- `ParamVecT bounded_theta_stack (const ParamVecT &theta, double epsilon=bounds_epsilon) const`
- `ParamVecT reflected_theta_stack (const ParamVecT &theta) const`
- `ImageT make_image () const`
- `ImageStackT make_image_stack (ImageCoordT n) const`
- `ImageCoordT get_size_image_stack (const ImageStackT &stack) const`
- `ImageT get_image_from_stack (const ImageStackT &stack, ImageCoordT n) const`
- `template<class ImT >`  
`void set_image_in_stack (ImageStackT &stack, ImageCoordT n, const ImT &im) const`
- `ImageSizeT get_size () const`
- `ImageCoordT get_size (IdxT idx) const`
- `ImageCoordT get_num_pixels () const`
- `void set_size (const ImageSizeT &size_)`
- `void set_size (const arma::Col< ImageCoordT > &sz)`
- `void check_image_shape (const ImageT &im) const`  
*Check the shape of a single images is correct for model size.*
- `void check_image_shape (const ImageStackT &ims) const`  
*Check the shape of a stack of images is correct for model size.*
- `void sample_mcmc_candidate (IdxT sample_index, ParamT &candidate, double step_scale=1.0) const`
- `void sample_mcmc_candidate (IdxT sample_index, ParamT &candidate, const IdxVecT &fixed_parameters_↔ mask, double step_scale=1.0) const`
- `void set_intensity_mcmc_sampling (double eta_l=-1)`
- `void set_background_mcmc_sampling (double eta_bg=-1)`
- `void set_mcmc_sigma_scale (double scale)`
- `double get_mcmc_sigma_scale () const`
- `IdxT get_mcmc_num_phases () const`

### Static Public Member Functions

- static CompositeDist [make\\_default\\_prior](#) (IdxT size, double min\_sigma, double max\_sigma, const std::string &prior\_type)
- static CompositeDist [make\\_default\\_prior\\_beta\\_position](#) (IdxT size, double min\_sigma, double max\_sigma)
- static CompositeDist [make\\_default\\_prior\\_normal\\_position](#) (IdxT size, double min\_sigma, double max\_sigma)
- static CompositeDist [make\\_prior\\_beta\\_position](#) (IdxT size, double beta\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double min\_sigma, double max\_sigma, double alpha\_sigma)
- static CompositeDist [make\\_prior\\_normal\\_position](#) (IdxT size, double sigma\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double min\_sigma, double max\_sigma, double alpha\_sigma)
- static prior\_hessian::TruncatedNormalDist [make\\_prior\\_component\\_position\\_normal](#) (IdxT size, double pos\_←, sigma=default\_sigma\_pos)
- static prior\_hessian::ScaledSymmetricBetaDist [make\\_prior\\_component\\_position\\_beta](#) (IdxT size, double pos\_←, beta=default\_beta\_pos)
- static prior\_hessian::TruncatedGammaDist [make\\_prior\\_component\\_intensity](#) (double mean=default\_mean\_←, l, double kappa=default\_intensity\_kappa)
- static prior\_hessian::TruncatedParetoDist [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_←, sigma, double alpha=default\_alpha\_sigma)
- static void [set\\_rng\\_seed](#) (RngSeedT seed)
- static ParallelRngManagerT & [get\\_rng\\_manager](#) ()
- static ParallelRngGeneratorT & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const ImageSizeT &size\_)

*Check the size argument for the model.*

### Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) = 1
- static const [ImageCoordT](#) [global\\_min\\_size](#) = 3
- static const [ImageCoordT](#) [global\\_max\\_size](#) = 512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5
- static const std::vector< std::string > [estimator\\_names](#)

### Protected Member Functions

- void [set\\_mcmc\\_num\\_phases](#) (IdxT num\_phases)



## Protected Attributes

- CompositeDist [prior](#)
- [IdxT](#) num\_params
- [IdxT](#) num\_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double [eta\\_sigma](#) = -1
- double [eta\\_x](#) = 0
- double [eta\\_l](#) = 0
- double [eta\\_bg](#) = 0
- [IdxT](#) num\_phases
- double [sigma\\_scale](#)

## 9.9.1 Detailed Description

A 1D Gaussian with variable PSF sigma under an Poisson read noise assumption and MAP Objective.

Model: [Gauss1DsModel](#) a 1D gaussian PSF with variable psf\_sigma Objective: [PoissonNoise1DObjective](#) - Poisson noise model for 1D Estimator: MAPstimator - Maximum a-posteriori estimator

Definition at line 24 of file Gauss1DsMAP.h.

## 9.9.2 Member Typedef Documentation

9.9.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [\[inherited\]](#)

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

9.9.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` [\[inherited\]](#)

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

9.9.2.3 `template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT>`  
[\[inherited\]](#)

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

**9.9.2.4** `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT` [inherited]

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

**9.9.2.5** `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

**9.9.2.6** `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT>`  
[inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

**9.9.2.7** `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
[inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

**9.9.2.8** `template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>`  
[inherited]

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

**9.9.2.9** `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
[inherited]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

**9.9.2.10** `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

9.9.2.11 `using mappel::PoissonNoise1DObjective::ModelDataStackT = ImageStackT` [inherited]

Objective function data stack type: 1D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 26 of file PoissonNoise1DObjective.h.

9.9.2.12 `using mappel::PoissonNoise1DObjective::ModelDataT = ImageT` [inherited]

Objective function data type: 1D double precision image, gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise1DObjective.h.

9.9.2.13 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.9.2.14 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

9.9.2.15 `using mappel::Gauss1DsModel::StencilVecT = std::vector<Stencil>` [inherited]

Definition at line 47 of file Gauss1DsModel.h.

### 9.9.3 Constructor & Destructor Documentation

9.9.3.1 `mappel::Gauss1DsMAP::Gauss1DsMAP ( arma::Col< ImageCoordT > size, VecT min_sigma, VecT max_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 12 of file Gauss1DsMAP.cpp.

9.9.3.2 `mappel::Gauss1DsMAP::Gauss1DsMAP ( ImageSizeT size, double min_sigma, double max_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 16 of file Gauss1DsMAP.cpp.

9.9.3.3 `mappel::Gauss1DsMAP::Gauss1DsMAP ( ImageSizeT size, CompositeDist && prior )`

Definition at line 20 of file Gauss1DsMAP.cpp.

#### 9.9.3.4 mappel::Gauss1DsMAP::Gauss1DsMAP ( ImageSizeT *size*, const CompositeDist & *prior* )

Definition at line 28 of file Gauss1DsMAP.cpp.

#### 9.9.3.5 mappel::Gauss1DsMAP::Gauss1DsMAP ( const Gauss1DsMAP & *o* )

Definition at line 36 of file Gauss1DsMAP.cpp.

#### 9.9.3.6 mappel::Gauss1DsMAP::Gauss1DsMAP ( Gauss1DsMAP && *o* )

Definition at line 44 of file Gauss1DsMAP.cpp.

### 9.9.4 Member Function Documentation

#### 9.9.4.1 void mappel::PointEmitterModel::bound\_theta ( ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

#### 9.9.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

#### 9.9.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

#### 9.9.4.4 void ImageFormat1DBase::check\_image\_shape ( const ImageT & *im* ) const [inherited]

Check the shape of a single images is correct for model size.

Definition at line 59 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.9.4.5 `void ImageFormat1DBase::check_image_shape ( const ImageStackT & ims ) const` [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.9.4.6 `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.9.4.7 `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

9.9.4.8 `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

9.9.4.9 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

9.9.4.10 `void ImageFormat1DBase::check_size ( const ImageSizeT & size )` [static], [inherited]

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::global\_max\_size, and mappel::ImageFormat1DBase::global\_min\_size.

Referenced by mappel::ImageFormat1DBase::ImageFormat1DBase(), and mappel::ImageFormat1DBase::set\_size().

**9.9.4.11** `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.9.4.12** `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` `[inline]`,`[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.9.4.13** `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.9.4.14** `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.9.4.15** `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` `[inline]`,`[inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

**9.9.4.16** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.9.4.17** `double mappel::Gauss1DsModel::get_max_sigma ( ) const` `[inline]`,`[inherited]`

Definition at line 102 of file Gauss1DsModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss1DsModel::get\_stats(), and mappel::Gauss1DsModel::set\_min\_sigma().

**9.9.4.18** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

**9.9.4.19** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

**9.9.4.20** `double mappel::Gauss1DsModel::get_min_sigma ( ) const` `[inline],[inherited]`

Definition at line 98 of file Gauss1DsModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss1DsModel::get_stats()`, and `mappel::Gauss1DsModel::set_max_sigma()`.

**9.9.4.21** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

**9.9.4.22** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.9.4.23** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_num_pixels ( ) const` `[inline],[inherited]`

Definition at line 82 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

**9.9.4.24** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

#### 9.9.4.25 `CompositeDist & mappel::PointEmitterModel::get_prior ( ) [inline],[inherited]`

Definition at line 200 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

#### 9.9.4.26 `const CompositeDist & mappel::PointEmitterModel::get_prior ( ) const [inline],[inherited]`

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

#### 9.9.4.27 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ( ) [static],[inherited]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

#### 9.9.4.28 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ( ) [static],[inherited]`

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

#### 9.9.4.29 `ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get_size ( ) const [inline],[inherited]`

Definition at line 71 of file `ImageFormat1DBase.h`.

References `mappel::ImageFormat1DBase::size`.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

#### 9.9.4.30 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size ( IdxT idx ) const [inherited]`

Definition at line 20 of file `ImageFormat1DBase.cpp`.

References `mappel::ImageFormat1DBase::size`.

#### 9.9.4.31 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack ( const ImageStackT & stack ) const [inline],[inherited]`

Definition at line 101 of file `ImageFormat1DBase.h`.



#### 9.9.4.32 StatsT mappel::Gauss1DsModel::get\_stats ( ) const [inherited]

Definition at line 198 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::MCMCAdaptor1Ds::get\_stats(), mappel::ImageFormat1DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

#### 9.9.4.33 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const [inline], [inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

#### 9.9.4.34 bool mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const [inline], [inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

#### 9.9.4.35 Gauss1DsModel::Stencil mappel::Gauss1DsModel::initial\_theta\_estimate ( const ImageT & im ) const [inline], [inherited]

Fast, heuristic estimate of initial theta.

Definition at line 173 of file Gauss1DsModel.h.

References mappel::PointEmitterModel::make\_param().

#### 9.9.4.36 Gauss1DsModel::Stencil mappel::Gauss1DsModel::initial\_theta\_estimate ( const ImageT & im, const ParamT & theta\_init ) const [inherited]

Definition at line 231 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::Stencil::bg(), mappel::Gauss1DsModel::Stencil::l(), mappel::PointEmitterModel::lbound, mappel::Gauss1DsModel::make\_stencil(), mappel::Gauss1DsModel::Stencil::sigma(), mappel::ImageFormat1DBase::size, and mappel::PointEmitterModel::ubound.

#### 9.9.4.37 CompositeDist mappel::Gauss1DsModel::make\_default\_prior ( IdxT size, double min\_sigma, double max\_sigma, const std::string & prior\_type ) [static], [inherited]

Definition at line 50 of file Gauss1DsModel.cpp.

References mappel::istarts\_with(), mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), and mappel::Gauss1DsModel::make\_default\_prior\_normal\_position().

**9.9.4.38 CompositeDist mappel::Gauss1DsModel::make\_default\_prior\_beta\_position ( IdxT size, double min\_sigma, double max\_sigma ) [static],[inherited]**

Definition at line 72 of file Gauss1DsModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component←\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), and mappel::PointEmitterModel←::make\_prior\_component\_sigma().

Referenced by mappel::Gauss1DsModel::make\_default\_prior().

**9.9.4.39 CompositeDist mappel::Gauss1DsModel::make\_default\_prior\_normal\_position ( IdxT size, double min\_sigma, double max\_sigma ) [static],[inherited]**

Definition at line 83 of file Gauss1DsModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component←\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), and mappel::PointEmitterModel←::make\_prior\_component\_sigma().

Referenced by mappel::Gauss1DsModel::make\_default\_prior().

**9.9.4.40 ImageFormat1DBase::ImageT ImageFormat1DBase::make\_image ( ) const [inline],[inherited]**

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.9.4.41 ImageFormat1DBase::ImageStackT ImageFormat1DBase::make\_image\_stack ( ImageCoordT n ) const [inline],[inherited]**

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.9.4.42 PointEmitterModel::ParamT mappel::PointEmitterModel::make\_param ( ) const [inline],[inherited]**

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::←Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel←::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess←update().

**9.9.4.43 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make\_param ( FillT fill ) const [inherited]**

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.9.4.44 **MatT** mappel::PointEmitterModel::make\_param\_mat ( ) const [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.9.4.45 **template<class FillT > MatT** mappel::PointEmitterModel::make\_param\_mat ( FillT fill ) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.9.4.46 **CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT n ) const [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.9.4.47 **template<class FillT > CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT n, FillT fill ) const [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.9.4.48 **PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make\_param\_stack ( IdxT n ) const [inline],[inherited]

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.9.4.49 **template<class FillT > PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make\_param\_stack ( IdxT n, FillT fill ) const [inherited]

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.9.4.50 **CompositeDist** mappel::Gauss1DsModel::make\_prior\_beta\_position ( IdxT size, double beta\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double min\_sigma, double max\_sigma, double alpha\_sigma ) [static],[inherited]

Definition at line 94 of file Gauss1DsModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), and mappel::PointEmitterModel::make\_prior\_component\_sigma().

Referenced by mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator().

**9.9.4.51** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_l, double kappa = default_intensity_kappa ) [static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.9.4.52** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos ) [static],[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

**9.9.4.53** `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos ) [static],[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.9.4.54** `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma ) [static],[inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.9.4.55 **CompositeDist** mappel::Gauss1DsModel::make\_prior\_normal\_position ( *IdxT size*, double *sigma\_xpos*, double *mean\_l*, double *kappa\_l*, double *mean\_bg*, double *kappa\_bg*, double *min\_sigma*, double *max\_sigma*, double *alpha\_sigma* )  
[static],[inherited]

Definition at line 108 of file Gauss1DsModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), and mappel::PointEmitterModel::make\_prior\_component\_sigma().

Referenced by mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator().

9.9.4.56 **Gauss1DsModel::Stencil** mappel::Gauss1DsModel::make\_stencil ( const ParamT & *theta*, bool *compute\_derivatives* =true ) const [inline],[inherited]

Make a new Model::Stencil object at theta.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular theta (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not model.theta\_in\_bounds(theta).

If derivatives will not be computed with this stencil set compute\_derivatives=false

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 123 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::Stencil(), and mappel::PointEmitterModel::theta\_in\_bounds().

Referenced by mappel::Gauss1DsModel::initial\_theta\_estimate().

9.9.4.57 **Gauss1DsMAP & mappel::Gauss1DsMAP::operator=** ( const Gauss1DsMAP & *o* )

Definition at line 52 of file Gauss1DsMAP.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise1DObjective::operator=(), mappel::Gauss1DsModel::operator=(), and mappel::PointEmitterModel::operator=().

#### 9.9.4.58 Gauss1DsMAP & mappel::Gauss1DsMAP::operator= ( Gauss1DsMAP && o )

Definition at line 63 of file Gauss1DsMAP.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise1DObjective::operator=(), mappel::Gauss1DsModel::operator=(), and mappel::PointEmitterModel::operator=().

#### 9.9.4.59 void mappel::Gauss1DsModel::pixel\_grad ( IdxT i, const Stencil & s, ParamT & pgrad ) const [inline], [inherited]

Definition at line 140 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::I(), and mappel::Gauss1DsModel::Stencil::X.

Referenced by mappel::Gauss1DsModel::pixel\_hess\_update().

#### 9.9.4.60 void mappel::Gauss1DsModel::pixel\_grad2 ( IdxT i, const Stencil & s, ParamT & pgrad2 ) const [inline], [inherited]

Definition at line 150 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::I(), and mappel::Gauss1DsModel::Stencil::sigma().

#### 9.9.4.61 void mappel::Gauss1DsModel::pixel\_hess ( IdxT i, const Stencil & s, MatT & hess ) const [inline], [inherited]

Definition at line 160 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::DXSX, mappel::Gauss1DsModel::Stencil::I(), and mappel::Gauss1DsModel::Stencil::sigma().

#### 9.9.4.62 void mappel::Gauss1DsModel::pixel\_hess\_update ( IdxT i, const Stencil & s, double dm\_ratio\_m1, double dmm\_ratio, ParamT & grad, MatT & hess ) const [inherited]

Definition at line 211 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::DXSX, mappel::Gauss1DsModel::Stencil::I(), mappel::PointEmitterModel::make\_param(), mappel::Gauss1DsModel::pixel\_grad(), and mappel::Gauss1DsModel::Stencil::sigma().

#### 9.9.4.63 double mappel::Gauss1DsModel::pixel\_model\_value ( IdxT i, const Stencil & s ) const [inline], [inherited]

Definition at line 134 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::bg(), mappel::Gauss1DsModel::Stencil::I(), and mappel::Gauss1DsModel::Stencil::X.

**9.9.4.64** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.9.4.65** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const`  
`[inherited]`

Definition at line 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.9.4.66** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.9.4.67** `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const`  
`[inherited]`

Definition at line 59 of file `MCMCAdaptor1Ds.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1Ds::eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.9.4.68** `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const`  
`[inherited]`

Definition at line 77 of file `MCMCAdaptor1Ds.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1Ds::eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.9.4.69** `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.9.4.70** `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline],[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

**9.9.4.71** `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` `[inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

**9.9.4.72** `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )` `[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.9.4.73** `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline],[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.9.4.74** `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline],[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.9.4.75** `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams )` `[inline],[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.



9.9.4.76 `template<class ImT > void ImageFormat1DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const` `[inherited]`

Definition at line 115 of file ImageFormat1DBase.h.

9.9.4.77 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 )` `[inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.9.4.78 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.9.4.79 `void mappel::Gauss1DsModel::set_max_sigma ( double max_sigma )` `[inherited]`

Definition at line 135 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::set_ubound()`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::Gauss2DsModel::set_min_sigma()`.

9.9.4.80 `void mappel::Gauss1DsModel::set_max_sigma ( const VecT & max_sigma )` `[inherited]`

Definition at line 153 of file Gauss1DsModel.cpp.

References `mappel::Gauss1DsModel::set_max_sigma()`.

9.9.4.81 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` `[protected]`, `[inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.9.4.82 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` `[inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma←_scale`.

9.9.4.83 `void mappel::Gauss1DsModel::set_min_sigma ( double min_sigma )` `[inherited]`

Definition at line 122 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::←PointEmitterModel::prior`, and `mappel::PointEmitterModel::set_lbound()`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`, and `mappel::Gauss2DsModel::set_min_sigma()`.

9.9.4.84 `void mappel::Gauss1DsModel::set_min_sigma ( const VecT & min_sigma )` `[inherited]`

Definition at line 148 of file Gauss1DsModel.cpp.

References `mappel::Gauss1DsModel::set_min_sigma()`.

9.9.4.85 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` `[inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.9.4.86 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )` `[inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::Point←EmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

9.9.4.87 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ )` `[inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::Point←EmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.9.4.88 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` `[static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.9.4.89 `void ImageFormat1DBase::set_size ( const ImageSizeT & size_ ) [inherited]`

Definition at line 30 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::check\_size(), and mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::set\_size(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DsModel::set\_size().

9.9.4.90 `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz ) [inline],[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::set\_size().

9.9.4.91 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_max\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.9.4.92 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::make\_stencil(), mappel::Gauss1DsModel::make\_stencil(), mappel::Gauss2DModel::make\_stencil(), mappel::Gauss2DModel::make\_stencil(), mappel::Gauss2DsxyModel::make\_stencil(), mappel::Gauss2DsModel::make\_stencil(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.9.4.93 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const [inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::theta\_in\_bounds().

## 9.9.5 Member Data Documentation

9.9.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static],[inherited]`

Distance from the boundary to constrain in bound\_theta and bounded\_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), and mappel::PointEmitterModel::set\_ubound().

**9.9.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static],[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

**9.9.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

**9.9.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.9.5.5** `const double mappel::PointEmitterModel::default_max_I = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.9.5.6** `const double mappel::PointEmitterModel::default_mean_I = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.9.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.9.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.9.5.9** `const std::string mappel::Gauss1DsModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 51 of file Gauss1DsModel.h.

Referenced by mappel::Gauss1DsModel::operator=().

**9.9.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), and mappel::Gauss2DsModel::initial\_theta\_estimate().

**9.9.5.11** `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names` `[static], [inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

**9.9.5.12** `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to theta\_bg in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by mappel::MCMCAdaptor1D::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1D::operator=(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.9.5.13** `double mappel::MCMCAdaptor1D::eta_I = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to theta\_I in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by mappel::MCMCAdaptor1D::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1D::operator=(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.9.5.14** `double mappel::MCMCAdaptor1Ds::eta_sigma = 1` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor1Ds.h`.

Referenced by `mappel::MCMCAdaptor1Ds::get_stats()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor1Ds::operator=()`, and `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`.

**9.9.5.15** `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

**9.9.5.16** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.9.5.17** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.9.5.18** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.9.5.19** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.9.5.20** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static]`, `[inherited]`

Global minimum for any psf\_sigma. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.9.5.21** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static]`, `[inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.9.5.22** `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.9.5.23** `const std::string mappel::Gauss1DsMAP::name` `[static]`

Definition at line 35 of file Gauss1DsMAP.h.

**9.9.5.24** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static]`, `[inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

**9.9.5.25** `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

#### 9.9.5.26 `IdxT mappel::PointEmitterModel::num_params` [protected],[inherited]

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.9.5.27 `IdxT mappel::MCMCAdaptorBase::num_phases` [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

#### 9.9.5.28 `CompositeDist mappel::PointEmitterModel::prior` [protected],[inherited]

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.9.5.29 `const StringVecT mappel::Gauss1DsModel::prior_types` [static],[inherited]

**Initial value:**

```
= { "Beta",
    "Normal"
}
```

Definition at line 50 of file `Gauss1DsModel.h`.

Referenced by `mappel::Gauss1DsModel::operator=()`.



### 9.9.5.30 double mappel::MCMCAdaptorBase::sigma\_scale [protected], [inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

### 9.9.5.31 ImageSizeT mappel::ImageFormat1DBase::size [protected], [inherited]

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::check\_image\_shape(), mappel::Gauss1DsModel::Stencil::compute\_derivatives(), mappel::Gauss1DModel::Stencil::compute\_derivatives(), mappel::ImageFormat1DBase::get\_num\_pixels(), mappel::ImageFormat1DBase::get\_size(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::ImageFormat1DBase::make\_image(), mappel::ImageFormat1DBase::make\_image\_stack(), mappel::ImageFormat1DBase::set\_size(), mappel::Gauss1DsModel::Stencil::Stencil(), and mappel::Gauss1DModel::Stencil::Stencil().

### 9.9.5.32 ParamT mappel::PointEmitterModel::ubound [protected], [inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

The documentation for this class was generated from the following files:

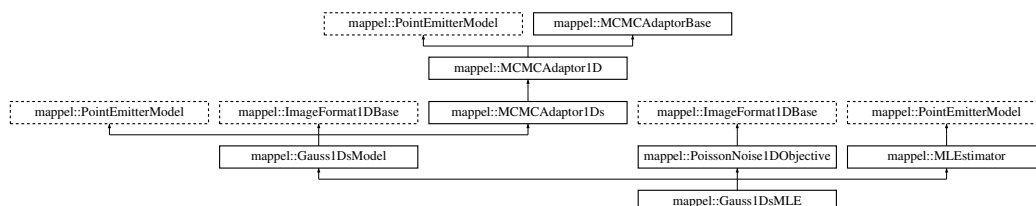
- [Gauss1DsMAP.h](#)
- [Gauss1DsMAP.cpp](#)

## 9.10 mappel::Gauss1DsMLE Class Reference

A 1D Gaussian with variable PSF under an Poisson noise assumption and maximum-likelihood estimator.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DsMLE.h>
```

Inheritance diagram for mappel::Gauss1DsMLE:



## Public Types

- using `StencilVecT` = `std::vector< Stencil >`
- using `ParamT` = `arma::vec`
- using `ParamVecT` = `arma::mat`
- using `ImageCoordT` = `uint32_t`
- using `ImagePixelT` = `double`
- template<class CoordT >  
using `ImageSizeShapeT` = `CoordT`
- template<class CoordT >  
using `ImageSizeVecShapeT` = `arma::Col< CoordT >`
- using `ImageSizeT` = `ImageSizeShapeT< ImageCoordT >`
- using `ImageSizeVecT` = `ImageSizeVecShapeT< ImageCoordT >`
- template<class PixelT >  
using `ImageShapeT` = `arma::Col< PixelT >`
- template<class PixelT >  
using `ImageStackShapeT` = `arma::Mat< PixelT >`
- using `ImageT` = `ImageShapeT< ImagePixelT >`
- using `ImageStackT` = `ImageStackShapeT< ImagePixelT >`
- using `ModelDataT` = `ImageT`
- using `ModelDataStackT` = `ImageStackT`

## Public Member Functions

- `Gauss1DsMLE` (`arma::Col< ImageCoordT > size`, `VecT min_sigma`, `VecT max_sigma`, `const std::string &prior_type=DefaultPriorType`)
- `Gauss1DsMLE` (`ImageSizeT size`, `double min_sigma`, `double max_sigma`, `const std::string &prior_type=DefaultPriorType`)
- `Gauss1DsMLE` (`ImageSizeT size`, `CompositeDist &&prior`)
- `Gauss1DsMLE` (`ImageSizeT size`, `const CompositeDist &prior`)
- `Gauss1DsMLE` (`const Gauss1DsMLE &o`)
- `Gauss1DsMLE & operator=` (`const Gauss1DsMLE &o`)
- `Gauss1DsMLE` (`Gauss1DsMLE &&o`)
- `Gauss1DsMLE & operator=` (`Gauss1DsMLE &&o`)
- `double get_min_sigma` () `const`
- `double get_max_sigma` () `const`
- `void set_min_sigma` (`double min_sigma`)
- `void set_min_sigma` (`const VecT &min_sigma`)
- `void set_max_sigma` (`double max_sigma`)
- `void set_max_sigma` (`const VecT &max_sigma`)
- `StatsT get_stats` () `const`
- `Stencil make_stencil` (`const ParamT &theta`, `bool compute_derivatives=true`) `const`  
*Make a new Model::Stencil object at theta.*
- `double pixel_model_value` (`IdxT i`, `const Stencil &s`) `const`
- `void pixel_grad` (`IdxT i`, `const Stencil &s`, `ParamT &pgrad`) `const`
- `void pixel_grad2` (`IdxT i`, `const Stencil &s`, `ParamT &pgrad2`) `const`
- `void pixel_hess` (`IdxT i`, `const Stencil &s`, `MatT &hess`) `const`
- `void pixel_hess_update` (`IdxT i`, `const Stencil &s`, `double dm_ratio_m1`, `double dmm_ratio`, `ParamT &grad`, `MatT &hess`) `const`
- `Stencil initial_theta_estimate` (`const ImageT &im`) `const`

*Fast, heuristic estimate of initial theta.*

- [Stencil initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init) const
- [IdxT get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT make\\_param](#) () const
- template<class FillT >  
  [ParamT make\\_param](#) (FillT fill) const
- [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT make\\_param\\_mat](#) () const
- template<class FillT >  
  [MatT make\\_param\\_mat](#) (FillT fill) const
- [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get\\_prior](#) ()
- const CompositeDist & [get\\_prior](#) () const
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- [IdxT get\\_num\\_hyperparams](#) () const
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- [VecT get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT sample\\_prior](#) (RngT &rng) const
- [ParamT sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get\\_lbound](#) () const
- const [ParamT](#) & [get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT make\\_image](#) () const

- `ImageStackT make_image_stack (ImageCoordT n) const`
- `ImageCoordT get_size_image_stack (const ImageStackT &stack) const`
- `ImageT get_image_from_stack (const ImageStackT &stack, ImageCoordT n) const`
- `template<class ImT >`  
`void set_image_in_stack (ImageStackT &stack, ImageCoordT n, const ImT &im) const`
- `ImageSizeT get_size () const`
- `ImageCoordT get_size (IdxT idx) const`
- `ImageCoordT get_num_pixels () const`
- `void set_size (const ImageSizeT &size_)`
- `void set_size (const arma::Col< ImageCoordT > &sz)`
- `void check_image_shape (const ImageT &im) const`  
*Check the shape of a single images is correct for model size.*
- `void check_image_shape (const ImageStackT &ims) const`  
*Check the shape of a stack of images is correct for model size.*
- `void sample_mcmc_candidate (IdxT sample_index, ParamT &candidate, double step_scale=1.0) const`
- `void sample_mcmc_candidate (IdxT sample_index, ParamT &candidate, const IdxVecT &fixed_parameters_↵  
mask, double step_scale=1.0) const`
- `void set_intensity_mcmc_sampling (double eta_l=-1)`
- `void set_background_mcmc_sampling (double eta_bg=-1)`
- `void set_mcmc_sigma_scale (double scale)`
- `double get_mcmc_sigma_scale () const`
- `IdxT get_mcmc_num_phases () const`

#### Static Public Member Functions

- static CompositeDist `make_default_prior (IdxT size, double min_sigma, double max_sigma, const std::string &prior_type)`
- static CompositeDist `make_default_prior_beta_position (IdxT size, double min_sigma, double max_sigma)`
- static CompositeDist `make_default_prior_normal_position (IdxT size, double min_sigma, double max_sigma)`
- static CompositeDist `make_prior_beta_position (IdxT size, double beta_xpos, double mean_l, double kappa_l,  
double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma)`
- static CompositeDist `make_prior_normal_position (IdxT size, double sigma_xpos, double mean_l, double  
kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma)`
- static prior\_hessian::TruncatedNormalDist `make_prior_component_position_normal (IdxT size, double pos_↵  
sigma=default_sigma_pos)`
- static prior\_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta (IdxT size, double pos_↵  
beta=default_beta_pos)`
- static prior\_hessian::TruncatedGammaDist `make_prior_component_intensity (double mean=default_mean_↵  
l, double kappa=default_intensity_kappa)`
- static prior\_hessian::TruncatedParetoDist `make_prior_component_sigma (double min_sigma, double max_↵  
sigma, double alpha=default_alpha_sigma)`
- static void `set_rng_seed (RngSeedT seed)`
- static `ParallelRngManagerT & get_rng_manager ()`
- static `ParallelRngGeneratorT & get_rng_generator ()`
- static void `check_size (const ImageSizeT &size_)`

*Check the size argument for the model.*

## Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) = 1
- static const [ImageCoordT](#) [global\\_min\\_size](#) = 3
- static const [ImageCoordT](#) [global\\_max\\_size](#) = 512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5
- static const std::vector< std::string > [estimator\\_names](#)

## Protected Member Functions

- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) [num\\_phases](#))

## Protected Attributes

- CompositeDist [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta\\_sigma](#) == -1
- double [eta\\_x](#) = 0
- double [eta\\_l](#) = 0
- double [eta\\_bg](#) = 0
- [IdxT](#) [num\\_phases](#)
- double [sigma\\_scale](#)

## 9.10.1 Detailed Description

A 1D Gaussian with variable PSF under an Poisson noise assumption and maximum-likelihood estimator.

Model: [Gauss1DsModel](#) - 1D Gaussian PSF with variable PSF sigma Objective: [PoissonNoise1DObjective](#) - Poisson noise model for 1D Estimator: [MLEstimator](#) - Pure-likelihood estimator

Definition at line 24 of file Gauss1DsMLE.h.

### 9.10.2 Member Typedef Documentation

#### 9.10.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

#### 9.10.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

#### 9.10.2.3 `template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT>` [inherited]

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

#### 9.10.2.4 `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT` [inherited]

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

#### 9.10.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

#### 9.10.2.6 `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT>` [inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

#### 9.10.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>` [inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

**9.10.2.8** `template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

**9.10.2.9** `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

**9.10.2.10** `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

**9.10.2.11** `using mappel::PoissonNoise1DObjective::ModelDataStackT = ImageStackT` `[inherited]`

Objective function data stack type: 1D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 26 of file PoissonNoise1DObjective.h.

**9.10.2.12** `using mappel::PoissonNoise1DObjective::ModelDataT = ImageT` `[inherited]`

Objective function data type: 1D double precision image, gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise1DObjective.h.

**9.10.2.13** `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

**9.10.2.14** `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

**9.10.2.15** `using mappel::Gauss1DsModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 47 of file Gauss1DsModel.h.

### 9.10.3 Constructor & Destructor Documentation

9.10.3.1 `mappel::Gauss1DsMLE::Gauss1DsMLE ( arma::Col< ImageCoordT > size, VecT min_sigma, VecT max_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 12 of file Gauss1DsMLE.cpp.

9.10.3.2 `mappel::Gauss1DsMLE::Gauss1DsMLE ( ImageSizeT size, double min_sigma, double max_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 16 of file Gauss1DsMLE.cpp.

9.10.3.3 `mappel::Gauss1DsMLE::Gauss1DsMLE ( ImageSizeT size, CompositeDist && prior )`

Definition at line 20 of file Gauss1DsMLE.cpp.

9.10.3.4 `mappel::Gauss1DsMLE::Gauss1DsMLE ( ImageSizeT size, const CompositeDist & prior )`

Definition at line 28 of file Gauss1DsMLE.cpp.

9.10.3.5 `mappel::Gauss1DsMLE::Gauss1DsMLE ( const Gauss1DsMLE & o )`

Definition at line 36 of file Gauss1DsMLE.cpp.

9.10.3.6 `mappel::Gauss1DsMLE::Gauss1DsMLE ( Gauss1DsMLE && o )`

Definition at line 44 of file Gauss1DsMLE.cpp.

### 9.10.4 Member Function Documentation

9.10.4.1 `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const`  
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

9.10.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.



**9.10.4.3** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

**9.10.4.4** `void ImageFormat1DBase::check_image_shape ( const ImageT & im ) const` [inherited]

Check the shape of a single images is correct for model size.

Definition at line 59 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::size`.

**9.10.4.5** `void ImageFormat1DBase::check_image_shape ( const ImageStackT & ims ) const` [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::size`.

**9.10.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.10.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

**9.10.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

9.10.4.9 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

9.10.4.10 `void ImageFormat1DBase::check_size ( const ImageSizeT & size_ )` [static],[inherited]

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::global_max_size`, and `mappel::ImageFormat1DBase::global_min_size`.

Referenced by `mappel::ImageFormat1DBase::ImageFormat1DBase()`, and `mappel::ImageFormat1DBase::set_size()`.

9.10.4.11 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline],[inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.10.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` [inline],[inherited]

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.10.4.13 `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.10.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` [inline],[inherited]

Definition at line 224 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.10.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` [inline],[inherited]

Definition at line 108 of file ImageFormat1DBase.h.

9.10.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

9.10.4.17 `double mappel::Gauss1DsModel::get_max_sigma ( ) const` `[inline]`,`[inherited]`

Definition at line 102 of file Gauss1DsModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss1DsModel::get\_stats(), and mappel::Gauss1DsModel::set\_min\_sigma().

9.10.4.18 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

9.10.4.19 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

9.10.4.20 `double mappel::Gauss1DsModel::get_min_sigma ( ) const` `[inline]`,`[inherited]`

Definition at line 98 of file Gauss1DsModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss1DsModel::get\_stats(), and mappel::Gauss1DsModel::set\_max\_sigma().

9.10.4.21 `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline]`,`[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

9.10.4.22 `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline]`,`[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.10.4.23 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get\_num\_pixels ( ) const** [inline], [inherited]

Definition at line 82 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.10.4.24 StringVecT mappel::PointEmitterModel::get\_param\_names ( ) const** [inline], [inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.10.4.25 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline], [inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.10.4.26 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline], [inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.10.4.27 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static], [inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.10.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( )** [static], [inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.10.4.29 ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get\_size ( ) const** [inline], [inherited]

Definition at line 71 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

9.10.4.30 **ImageFormat1DBase::ImageCoordT** ImageFormat1DBase::get\_size ( *IdxT idx* ) const [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.10.4.31 **ImageFormat1DBase::ImageCoordT** ImageFormat1DBase::get\_size\_image\_stack ( const ImageStackT & *stack* ) const [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

9.10.4.32 **StatsT** mappel::Gauss1DsModel::get\_stats ( ) const [inherited]

Definition at line 198 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::MCMCAdaptor1Ds::get\_stats(), mappel::ImageFormat1DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

9.10.4.33 **const PointEmitterModel::ParamT &** mappel::PointEmitterModel::get\_ubound ( ) const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.10.4.34 **bool** mappel::PointEmitterModel::has\_hyperparam ( const std::string & *name* ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.10.4.35 **Gauss1DsModel::Stencil** mappel::Gauss1DsModel::initial\_theta\_estimate ( const ImageT & *im* ) const [inline],[inherited]

Fast, heuristic estimate of initial theta.

Definition at line 173 of file Gauss1DsModel.h.

References mappel::PointEmitterModel::make\_param().

9.10.4.36 **Gauss1DsModel::Stencil** mappel::Gauss1DsModel::initial\_theta\_estimate ( const ImageT & *im*, const ParamT & *theta\_init* ) const [inherited]

Definition at line 231 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::Stencil::bg(), mappel::Gauss1DsModel::Stencil::l(), mappel::PointEmitterModel::lbound, mappel::Gauss1DsModel::make\_stencil(), mappel::Gauss1DsModel::Stencil::sigma(), mappel::ImageFormat1DBase::size, and mappel::PointEmitterModel::ubound.

**9.10.4.37 CompositeDist mappel::Gauss1DsModel::make\_default\_prior ( IdxT size, double min\_sigma, double max\_sigma, const std::string & prior\_type ) [static],[inherited]**

Definition at line 50 of file Gauss1DsModel.cpp.

References mappel::istarts\_with(), mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), and mappel::Gauss1DsModel::make\_default\_prior\_normal\_position().

**9.10.4.38 CompositeDist mappel::Gauss1DsModel::make\_default\_prior\_beta\_position ( IdxT size, double min\_sigma, double max\_sigma ) [static],[inherited]**

Definition at line 72 of file Gauss1DsModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), and mappel::PointEmitterModel::make\_prior\_component\_sigma().

Referenced by mappel::Gauss1DsModel::make\_default\_prior().

**9.10.4.39 CompositeDist mappel::Gauss1DsModel::make\_default\_prior\_normal\_position ( IdxT size, double min\_sigma, double max\_sigma ) [static],[inherited]**

Definition at line 83 of file Gauss1DsModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), and mappel::PointEmitterModel::make\_prior\_component\_sigma().

Referenced by mappel::Gauss1DsModel::make\_default\_prior().

**9.10.4.40 ImageFormat1DBase::ImageT ImageFormat1DBase::make\_image ( ) const [inline],[inherited]**

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.10.4.41 ImageFormat1DBase::ImageStackT ImageFormat1DBase::make\_image\_stack ( ImageCoordT n ) const [inline],[inherited]**

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.10.4.42 PointEmitterModel::ParamT mappel::PointEmitterModel::make\_param ( ) const [inline],[inherited]**

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

9.10.4.43 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const`  
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.10.4.44 `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.10.4.45 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.10.4.46 `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.10.4.47 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`  
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.10.4.48 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.10.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.10.4.50** `CompositeDist mappel::Gauss1DsModel::make_prior_beta_position ( IdxT size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma )` [static], [inherited]

Definition at line 94 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::PointEmitterModel::make_prior_component_sigma()`.

Referenced by `mappel::Gauss2DsModel::make_internal_1Dsum_estimator()`.

**9.10.4.51** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_l, double kappa = default_intensity_kappa )` [static], [inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.10.4.52** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos )` [static], [inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

**9.10.4.53** `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos )` [static], [inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.



9.10.4.54 **prior\_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make\_prior\_component\_sigma ( double *min\_sigma*, double *max\_sigma*, double *alpha* = default\_alpha\_sigma ) [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.10.4.55 **CompositeDist** mappel::Gauss1DsModel::make\_prior\_normal\_position ( IdxT *size*, double *sigma\_xpos*, double *mean\_l*, double *kappa\_l*, double *mean\_bg*, double *kappa\_bg*, double *min\_sigma*, double *max\_sigma*, double *alpha\_sigma* ) [static],[inherited]

Definition at line 108 of file Gauss1DsModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), and mappel::PointEmitterModel::make\_prior\_component\_sigma().

Referenced by mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator().

9.10.4.56 **Gauss1DsModel::Stencil** mappel::Gauss1DsModel::make\_stencil ( const ParamT & *theta*, bool *compute\_derivatives* =true ) const [inline],[inherited]

Make a new Model::Stencil object at *theta*.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular *theta* (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not model.theta\_in\_bounds(*theta*).

If derivatives will not be computed with this stencil set *compute\_derivatives*=false

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 123 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::Stencil(), and mappel::PointEmitterModel::theta\_in\_bounds().

Referenced by mappel::Gauss1DsModel::initial\_theta\_estimate().

#### 9.10.4.57 **Gauss1DsMLE & mappel::Gauss1DsMLE::operator= ( const Gauss1DsMLE & o )**

Definition at line 52 of file Gauss1DsMLE.cpp.

References mappel::MLEstimator::operator=(), mappel::PoissonNoise1DObjective::operator=(), mappel::Gauss1DsModel::operator=(), and mappel::PointEmitterModel::operator=().

#### 9.10.4.58 **Gauss1DsMLE & mappel::Gauss1DsMLE::operator= ( Gauss1DsMLE && o )**

Definition at line 63 of file Gauss1DsMLE.cpp.

References mappel::MLEstimator::operator=(), mappel::PoissonNoise1DObjective::operator=(), mappel::Gauss1DsModel::operator=(), and mappel::PointEmitterModel::operator=().

#### 9.10.4.59 **void mappel::Gauss1DsModel::pixel\_grad ( IdxT i, const Stencil & s, ParamT & pgrad ) const [inline], [inherited]**

Definition at line 140 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::l(), and mappel::Gauss1DsModel::Stencil::X.

Referenced by mappel::Gauss1DsModel::pixel\_hess\_update().

#### 9.10.4.60 **void mappel::Gauss1DsModel::pixel\_grad2 ( IdxT i, const Stencil & s, ParamT & pgrad2 ) const [inline], [inherited]**

Definition at line 150 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::l(), and mappel::Gauss1DsModel::Stencil::sigma().

#### 9.10.4.61 **void mappel::Gauss1DsModel::pixel\_hess ( IdxT i, const Stencil & s, MatT & hess ) const [inline], [inherited]**

Definition at line 160 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::DXSX, mappel::Gauss1DsModel::Stencil::l(), and mappel::Gauss1DsModel::Stencil::sigma().

#### 9.10.4.62 **void mappel::Gauss1DsModel::pixel\_hess\_update ( IdxT i, const Stencil & s, double dm\_ratio\_m1, double dmm\_ratio, ParamT & grad, MatT & hess ) const [inherited]**

Definition at line 211 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::DXSX, mappel::Gauss1DsModel::Stencil::l(), mappel::PointEmitterModel::make\_param(), mappel::Gauss1DsModel::pixel\_grad(), and mappel::Gauss1DsModel::Stencil::sigma().

**9.10.4.63** `double mappel::Gauss1DsModel::pixel_model_value ( IdxT i, const Stencil & s ) const` `[inline]`,  
`[inherited]`

Definition at line 134 of file Gauss1DsModel.h.

References `mappel::Gauss1DsModel::Stencil::bg()`, `mappel::Gauss1DsModel::Stencil::l()`, and `mappel::Gauss1DsModel::Stencil::X`.

**9.10.4.64** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.10.4.65** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.10.4.66** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.10.4.67** `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 59 of file MCMCAdaptor1Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1Ds::eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.10.4.68** `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 77 of file MCMCAdaptor1Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1Ds::eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.10.4.69 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.10.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline]`, `[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.10.4.71 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` `[inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.10.4.72 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.10.4.73 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline]`,  
`[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.10.4.74 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline]`,  
`[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.10.4.75 `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams ) [inline], [inherited]`

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

9.10.4.76 `template<class ImT > void ImageFormat1DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]`

Definition at line 115 of file ImageFormat1DBase.h.

9.10.4.77 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 ) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.10.4.78 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.10.4.79 `void mappel::Gauss1DsModel::set_max_sigma ( double max_sigma ) [inherited]`

Definition at line 135 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::set_ubound()`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::Gauss2DsModel::set_min_sigma()`.

9.10.4.80 `void mappel::Gauss1DsModel::set_max_sigma ( const VecT & max_sigma ) [inherited]`

Definition at line 153 of file Gauss1DsModel.cpp.

References `mappel::Gauss1DsModel::set_max_sigma()`.

9.10.4.81 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` [protected],[inherited]

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.10.4.82 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` [inherited]

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.10.4.83 `void mappel::Gauss1DsModel::set_min_sigma ( double min_sigma )` [inherited]

Definition at line 122 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::set_lbound()`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`, and `mappel::Gauss2DsModel::set_min_sigma()`.

9.10.4.84 `void mappel::Gauss1DsModel::set_min_sigma ( const VecT & min_sigma )` [inherited]

Definition at line 148 of file Gauss1DsModel.cpp.

References `mappel::Gauss1DsModel::set_min_sigma()`.

9.10.4.85 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.10.4.86 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior )` [inherited]

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

9.10.4.87 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior )` [inherited]

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.10.4.88 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.10.4.89 `void ImageFormat1DBase::set_size ( const ImageSizeT & size_ ) [inherited]`

Definition at line 30 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::check\_size(), and mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::set\_size(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DsModel::set\_size().

9.10.4.90 `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz ) [inline],[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::set\_size().

9.10.4.91 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_max\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.10.4.92 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::make\_stencil(), mappel::Gauss1DsModel::make\_stencil(), mappel::Gauss2DModel::make\_stencil(), mappel::Gauss2DsxyModel::make\_stencil(), mappel::Gauss2DsModel::make\_stencil(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.10.4.93 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const [inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::theta\_in\_bounds().

### 9.10.5 Member Data Documentation

**9.10.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

**9.10.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file `PointEmitterModel.h`.

**9.10.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file `PointEmitterModel.h`.

**9.10.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file `PointEmitterModel.h`.

**9.10.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.10.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file `PointEmitterModel.h`.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.



9.10.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.10.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

9.10.5.9 `const std::string mappel::Gauss1DsModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 51 of file Gauss1DsModel.h.

Referenced by `mappel::Gauss1DsModel::operator=()`.

9.10.5.10 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.10.5.11 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names` `[static], [inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

9.10.5.12 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

#### 9.10.5.13 `double mappel::MCMCAdaptor1D::eta_l=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

#### 9.10.5.14 `double mappel::MCMCAdaptor1Ds::eta_sigma=-1` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor1Ds.h`.

Referenced by `mappel::MCMCAdaptor1Ds::get_stats()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor1Ds::operator=()`, and `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`.

#### 9.10.5.15 `double mappel::MCMCAdaptor1D::eta_x=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

#### 9.10.5.16 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static],[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

#### 9.10.5.17 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static],[inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

#### 9.10.5.18 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static],[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.10.5.19** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static]`,  
`[inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::check\_size().

**9.10.5.20** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static]`, `[inherited]`

Global minimum for any psf\_sigma. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::check\_psf\_sigma().

**9.10.5.21** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static]`,  
`[inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::check\_size().

**9.10.5.22** `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

**9.10.5.23** `const std::string mappel::Gauss1DsMLE::name` `[static]`

Definition at line 35 of file Gauss1DsMLE.h.

**9.10.5.24** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static]`, `[inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::get\_stats().

#### 9.10.5.25 **IdxT** mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.10.5.26 **IdxT** mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.10.5.27 **IdxT** mappel::MCMCAdaptorBase::num\_phases [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().

#### 9.10.5.28 **CompositeDist** mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

#### 9.10.5.29 const StringVecT mappel::Gauss1DsModel::prior\_types [static],[inherited]

**Initial value:**

```
= { "Beta",  
                                     "Normal"  
}
```

Definition at line 50 of file Gauss1DsModel.h.

Referenced by mappel::Gauss1DsModel::operator=().

#### 9.10.5.30 double mappel::MCMCAdaptorBase::sigma\_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

#### 9.10.5.31 ImageSizeT mappel::ImageFormat1DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::check\_image\_shape(), mappel::Gauss1DsModel::Stencil::compute\_derivatives(), mappel::Gauss1DModel::Stencil::compute\_derivatives(), mappel::ImageFormat1DBase::get\_num\_pixels(), mappel::ImageFormat1DBase::get\_size(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::ImageFormat1DBase::make\_image(), mappel::ImageFormat1DBase::make\_image\_stack(), mappel::ImageFormat1DBase::set\_size(), mappel::Gauss1DsModel::Stencil::Stencil(), and mappel::Gauss1DModel::Stencil::Stencil().

#### 9.10.5.32 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

The documentation for this class was generated from the following files:

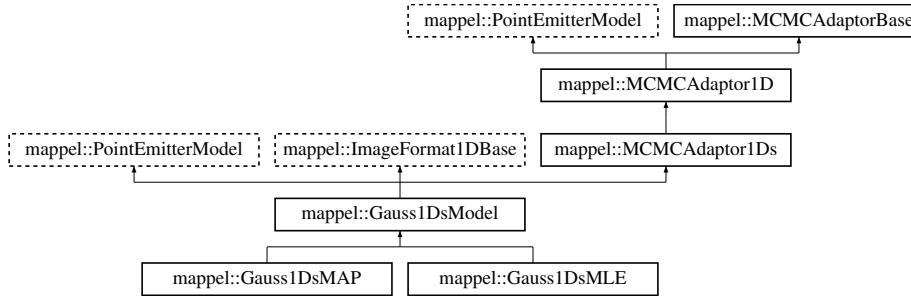
- [Gauss1DsMLE.h](#)
- [Gauss1DsMLE.cpp](#)

## 9.11 mappel::Gauss1DsModel Class Reference

Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DsModel.h>
```

Inheritance diagram for mappel::Gauss1DsModel:



### Classes

- class [Stencil](#)  
*Stencil for 1D variable-sigma models.*

### Public Types

- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = CoordT
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Col< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Col< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Mat< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >

## Public Member Functions

- double [get\\_min\\_sigma](#) () const
- double [get\\_max\\_sigma](#) () const
- void [set\\_min\\_sigma](#) (double min\_sigma)
- void [set\\_max\\_sigma](#) (double max\_sigma)
- void [set\\_min\\_sigma](#) (const [VecT](#) &min\_sigma)
- void [set\\_max\\_sigma](#) (const [VecT](#) &max\_sigma)
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &theta, bool compute\_derivatives=true) const  
*Make a new Model::Stencil object at theta.*
- double [pixel\\_model\\_value](#) ([IdxT](#) i, const [Stencil](#) &s) const
- void [pixel\\_grad](#) ([IdxT](#) i, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel\\_grad2](#) ([IdxT](#) i, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel\\_hess](#) ([IdxT](#) i, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel\\_hess\\_update](#) ([IdxT](#) i, const [Stencil](#) &s, double dm\_ratio\_m1, double dmm\_ratio, [ParamT](#) &grad, [MatT](#) &hess) const
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im) const  
*Fast, heuristic estimate of initial theta.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init) const
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT](#) [make\\_param](#) () const
- template<class FillT >  
[ParamT](#) [make\\_param](#) (FillT fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
[ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT](#) [make\\_param\\_mat](#) () const
- template<class FillT >  
[MatT](#) [make\\_param\\_mat](#) (FillT fill) const
- [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
[CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- [CompositeDist](#) & [get\\_prior](#) ()
- const [CompositeDist](#) & [get\\_prior](#) () const
- void [set\\_prior](#) ([CompositeDist](#) &&prior\_)
- void [set\\_prior](#) (const [CompositeDist](#) &prior\_)
- [IdxT](#) [get\\_num\\_hyperparams](#) () const
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- [VecT](#) [get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT](#) [get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)

- [StringVecT get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT sample\\_prior](#) (RngT &rng) const
- [ParamT sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) &[get\\_lbound](#) () const
- const [ParamT](#) &[get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT make\\_image](#) () const
- [ImageStackT make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- template<class ImT >  
  void [set\\_image\\_in\\_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const ImT &im) const
- [ImageSizeT get\\_size](#) () const
- [ImageCoordT get\\_size](#) ([IdxT](#) idx) const
- [ImageCoordT get\\_num\\_pixels](#) () const
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- void [set\\_size](#) (const arma::Col< [ImageCoordT](#) > &sz)
- void [check\\_image\\_shape](#) (const [ImageT](#) &im) const  
  *Check the shape of a single images is correct for model size.*
- void [check\\_image\\_shape](#) (const [ImageStackT](#) &ims) const  
  *Check the shape of a stack of images is correct for model size.*
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, const [IdxVecT](#) &fixed\_parameters\_↵  
  mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)
- void [set\\_background\\_mcmc\\_sampling](#) (double eta\_bg=-1)
- void [set\\_mcmc\\_sigma\\_scale](#) (double scale)
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- [IdxT get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Member Functions

- static CompositeDist [make\\_default\\_prior](#) ([IdxT](#) size, double min\_sigma, double max\_sigma, const std::string &prior\_type)
- static CompositeDist [make\\_default\\_prior\\_beta\\_position](#) ([IdxT](#) size, double min\_sigma, double max\_sigma)
- static CompositeDist [make\\_default\\_prior\\_normal\\_position](#) ([IdxT](#) size, double min\_sigma, double max\_sigma)
- static CompositeDist [make\\_prior\\_beta\\_position](#) ([IdxT](#) size, double beta\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double min\_sigma, double max\_sigma, double alpha\_sigma)



- static CompositeDist [make\\_prior\\_normal\\_position](#) (IdxT size, double sigma\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double min\_sigma, double max\_sigma, double alpha\_sigma)
- static prior\_hessian::TruncatedNormalDist [make\\_prior\\_component\\_position\\_normal](#) (IdxT size, double pos\_↵ sigma=default\_sigma\_pos)
- static prior\_hessian::ScaledSymmetricBetaDist [make\\_prior\\_component\\_position\\_beta](#) (IdxT size, double pos\_↵ beta=default\_beta\_pos)
- static prior\_hessian::TruncatedGammaDist [make\\_prior\\_component\\_intensity](#) (double mean=default\_mean\_↵ l, double kappa=default\_intensity\_kappa)
- static prior\_hessian::TruncatedParetoDist [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵ sigma, double alpha=default\_alpha\_sigma)
- static void [set\\_rng\\_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const ImageSizeT &size\_)

*Check the size argument for the model.*

#### Static Public Attributes

- static const StringVecT [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const ImageCoordT [num\\_dim](#) = 1
- static const ImageCoordT [global\\_min\\_size](#) = 3
- static const ImageCoordT [global\\_max\\_size](#) = 512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5

#### Protected Member Functions

- [Gauss1DsModel](#) (IdxT size\_)
- [Gauss1DsModel](#) (const Gauss1DsModel &o)
- [Gauss1DsModel](#) (Gauss1DsModel &&o)
- [Gauss1DsModel](#) & [operator=](#) (const Gauss1DsModel &o)
- [Gauss1DsModel](#) & [operator=](#) (Gauss1DsModel &&o)
- void [set\\_mcmc\\_num\\_phases](#) (IdxT num\_phases)

### Protected Attributes

- CompositeDist [prior](#)
- [IdxT](#) num\_params
- [IdxT](#) num\_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double [eta\\_sigma](#) ==-1
- double [eta\\_x](#) =0
- double [eta\\_l](#) =0
- double [eta\\_bg](#) =0
- [IdxT](#) num\_phases
- double [sigma\\_scale](#)

#### 9.11.1 Detailed Description

Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels.

Definition at line 19 of file Gauss1DsModel.h.

#### 9.11.2 Member Typedef Documentation

##### 9.11.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` `[inherited]`

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

##### 9.11.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

##### 9.11.2.3 `template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT>` `[inherited]`

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

##### 9.11.2.4 `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT` `[inherited]`

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

**9.11.2.5** `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`  
`[inherited]`

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

**9.11.2.6** `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT>`  
`[inherited]`

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

**9.11.2.7** `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
`[inherited]`

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

**9.11.2.8** `template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

**9.11.2.9** `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

**9.11.2.10** `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

**9.11.2.11** `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.11.2.12 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file `PointEmitterModel.h`.

9.11.2.13 `using mappel::Gauss1DsModel::StencilVecT = std::vector<Stencil>`

Definition at line 47 of file `Gauss1DsModel.h`.

### 9.11.3 Constructor & Destructor Documentation

9.11.3.1 `mappel::Gauss1DsModel::Gauss1DsModel ( IdxT size_ )` `[explicit],[protected]`

Definition at line 12 of file `Gauss1DsModel.cpp`.

9.11.3.2 `mappel::Gauss1DsModel::Gauss1DsModel ( const Gauss1DsModel & o )` `[protected]`

Definition at line 17 of file `Gauss1DsModel.cpp`.

9.11.3.3 `mappel::Gauss1DsModel::Gauss1DsModel ( Gauss1DsModel && o )` `[protected]`

Definition at line 22 of file `Gauss1DsModel.cpp`.

### 9.11.4 Member Function Documentation

9.11.4.1 `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const`  
`[inherited]`

Definition at line 248 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

9.11.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` `[inherited]`

Definition at line 265 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

9.11.4.3 **PointEmitterModel::ParamVecT** mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

9.11.4.4 **void** ImageFormat1DBase::check\_image\_shape ( const ImageT & *im* ) const [inherited]

Check the shape of a single images is correct for model size.

Definition at line 59 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.11.4.5 **void** ImageFormat1DBase::check\_image\_shape ( const ImageStackT & *ims* ) const [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.11.4.6 **void** mappel::PointEmitterModel::check\_param\_shape ( const ParamT & *theta* ) const [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.11.4.7 **void** mappel::PointEmitterModel::check\_param\_shape ( const ParamVecT & *theta* ) const [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

9.11.4.8 **void** mappel::PointEmitterModel::check\_psf\_sigma ( double *psf\_sigma* ) const [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

9.11.4.9 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

9.11.4.10 `void ImageFormat1DBase::check_size ( const ImageSizeT & size_ )` [static],[inherited]

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::global_max_size`, and `mappel::ImageFormat1DBase::global_min_size`.

Referenced by `mappel::ImageFormat1DBase::ImageFormat1DBase()`, and `mappel::ImageFormat1DBase::set_size()`.

9.11.4.11 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline],[inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.11.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` [inline],[inherited]

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.11.4.13 `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.11.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` [inline],[inherited]

Definition at line 224 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.11.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` [inline],[inherited]

Definition at line 108 of file ImageFormat1DBase.h.

9.11.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

9.11.4.17 `double mappel::Gauss1DsModel::get_max_sigma ( ) const` `[inline]`

Definition at line 102 of file Gauss1DsModel.h.

References mappel::PointEmitterModel::prior.

Referenced by get\_stats(), and set\_min\_sigma().

9.11.4.18 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

9.11.4.19 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

9.11.4.20 `double mappel::Gauss1DsModel::get_min_sigma ( ) const` `[inline]`

Definition at line 98 of file Gauss1DsModel.h.

References mappel::PointEmitterModel::prior.

Referenced by get\_stats(), and set\_max\_sigma().

9.11.4.21 `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline]`, `[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

9.11.4.22 `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline]`, `[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.11.4.23 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get\_num\_pixels ( ) const** [inline], [inherited]

Definition at line 82 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.11.4.24 StringVecT mappel::PointEmitterModel::get\_param\_names ( ) const** [inline], [inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.11.4.25 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline], [inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.11.4.26 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline], [inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.11.4.27 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static], [inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.11.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( )** [static], [inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.11.4.29 ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get\_size ( ) const** [inline], [inherited]

Definition at line 71 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().



9.11.4.30 **ImageFormat1DBase::ImageCoordT** ImageFormat1DBase::get\_size ( *IdxT idx* ) const [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.11.4.31 **ImageFormat1DBase::ImageCoordT** ImageFormat1DBase::get\_size\_image\_stack ( const ImageStackT & *stack* ) const [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

9.11.4.32 **StatsT** mappel::Gauss1DsModel::get\_stats ( ) const

Definition at line 198 of file Gauss1DsModel.cpp.

References get\_max\_sigma(), get\_min\_sigma(), mappel::MCMCAdaptor1Ds::get\_stats(), mappel::ImageFormat1DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

9.11.4.33 **const PointEmitterModel::ParamT &** mappel::PointEmitterModel::get\_ubound ( ) const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.11.4.34 **bool** mappel::PointEmitterModel::has\_hyperparam ( const std::string & *name* ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.11.4.35 **Gauss1DsModel::Stencil** mappel::Gauss1DsModel::initial\_theta\_estimate ( const ImageT & *im* ) const [inline]

Fast, heuristic estimate of initial theta.

Definition at line 173 of file Gauss1DsModel.h.

References mappel::PointEmitterModel::make\_param().

9.11.4.36 **Gauss1DsModel::Stencil** mappel::Gauss1DsModel::initial\_theta\_estimate ( const ImageT & *im*, const ParamT & *theta\_init* ) const

Definition at line 231 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::Stencil::bg(), mappel::Gauss1DsModel::Stencil::l(), mappel::PointEmitterModel::lbound, make\_stencil(), mappel::Gauss1DsModel::Stencil::sigma(), mappel::ImageFormat1DBase::size, and mappel::PointEmitterModel::ubound.

**9.11.4.37 CompositeDist** `mappel::Gauss1DsModel::make_default_prior ( IdxT size, double min_sigma, double max_sigma, const std::string & prior_type ) [static]`

Definition at line 50 of file Gauss1DsModel.cpp.

References `mappel::istarts_with()`, `make_default_prior_beta_position()`, and `make_default_prior_normal_position()`.

**9.11.4.38 CompositeDist** `mappel::Gauss1DsModel::make_default_prior_beta_position ( IdxT size, double min_sigma, double max_sigma ) [static]`

Definition at line 72 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::PointEmitterModel::make_prior_component_sigma()`.

Referenced by `make_default_prior()`.

**9.11.4.39 CompositeDist** `mappel::Gauss1DsModel::make_default_prior_normal_position ( IdxT size, double min_sigma, double max_sigma ) [static]`

Definition at line 83 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::PointEmitterModel::make_prior_component_sigma()`.

Referenced by `make_default_prior()`.

**9.11.4.40 ImageFormat1DBase::ImageT** `ImageFormat1DBase::make_image ( ) const [inline],[inherited]`

Definition at line 87 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

**9.11.4.41 ImageFormat1DBase::ImageStackT** `ImageFormat1DBase::make_image_stack ( ImageCoordT n ) const [inline],[inherited]`

Definition at line 94 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

**9.11.4.42 PointEmitterModel::ParamT** `mappel::PointEmitterModel::make_param ( ) const [inline],[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

9.11.4.43 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const`  
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.11.4.44 `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.11.4.45 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.11.4.46 `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.11.4.47 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`  
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.11.4.48 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.11.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.11.4.50 **CompositeDist** `mappel::Gauss1DsModel::make_prior_beta_position ( IdxT size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma ) [static]`

Definition at line 94 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::PointEmitterModel::make_prior_component_sigma()`.

Referenced by `mappel::Gauss2DsModel::make_internal_1Dsum_estimator()`.

9.11.4.51 **prior\_hessian::TruncatedGammaDist** `mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_l, double kappa = default_intensity_kappa ) [static], [inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.11.4.52 **prior\_hessian::ScaledSymmetricBetaDist** `mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos ) [static], [inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

9.11.4.53 **prior\_hessian::TruncatedNormalDist** `mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos ) [static], [inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.11.4.54 **prior\_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make\_prior\_component\_sigma ( double *min\_sigma*, double *max\_sigma*, double *alpha* = default\_alpha\_sigma ) [static], [inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.11.4.55 **CompositeDist** mappel::Gauss1DsModel::make\_prior\_normal\_position ( IdxT *size*, double *sigma\_xpos*, double *mean\_l*, double *kappa\_l*, double *mean\_bg*, double *kappa\_bg*, double *min\_sigma*, double *max\_sigma*, double *alpha\_sigma* ) [static]

Definition at line 108 of file Gauss1DsModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), and mappel::PointEmitterModel::make\_prior\_component\_sigma().

Referenced by mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator().

9.11.4.56 **Gauss1DsModel::Stencil** mappel::Gauss1DsModel::make\_stencil ( const ParamT & *theta*, bool *compute\_derivatives* =true ) const [inline]

Make a new Model::Stencil object at theta.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular theta (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not model.theta\_in\_bounds(theta).

If derivatives will not be computed with this stencil set compute\_derivatives=false

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 123 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::Stencil(), and mappel::PointEmitterModel::theta\_in\_bounds().

Referenced by initial\_theta\_estimate().

#### 9.11.4.57 **Gauss1DsModel & mappel::Gauss1DsModel::operator= ( const Gauss1DsModel & o )** [protected]

Definition at line 27 of file Gauss1DsModel.cpp.

References mappel::MCMCAdaptor1Ds::operator=().

Referenced by mappel::Gauss1DsMAP::operator=(), and mappel::Gauss1DsMLE::operator=().

#### 9.11.4.58 **Gauss1DsModel & mappel::Gauss1DsModel::operator= ( Gauss1DsModel && o )** [protected]

Definition at line 35 of file Gauss1DsModel.cpp.

References DefaultPriorType, mappel::MCMCAdaptor1Ds::operator=(), and prior\_types.

#### 9.11.4.59 **void mappel::Gauss1DsModel::pixel\_grad ( IdxT i, const Stencil & s, ParamT & pgrad ) const** [inline]

Definition at line 140 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::l(), and mappel::Gauss1DsModel::Stencil::X.

Referenced by pixel\_hess\_update().

#### 9.11.4.60 **void mappel::Gauss1DsModel::pixel\_grad2 ( IdxT i, const Stencil & s, ParamT & pgrad2 ) const** [inline]

Definition at line 150 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::l(), and mappel::Gauss1DsModel::Stencil::sigma().

#### 9.11.4.61 **void mappel::Gauss1DsModel::pixel\_hess ( IdxT i, const Stencil & s, MatT & hess ) const** [inline]

Definition at line 160 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::DXSX, mappel::Gauss1DsModel::Stencil::l(), and mappel::Gauss1DsModel::Stencil::sigma().

#### 9.11.4.62 **void mappel::Gauss1DsModel::pixel\_hess\_update ( IdxT i, const Stencil & s, double dm\_ratio\_m1, double dmm\_ratio, ParamT & grad, MatT & hess ) const**

Definition at line 211 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::DXSX, mappel::Gauss1DsModel::Stencil::l(), mappel::PointEmitterModel::make\_param(), pixel\_grad(), and mappel::Gauss1DsModel::Stencil::sigma().

9.11.4.63 `double mappel::Gauss1DsModel::pixel_model_value ( IdxT i, const Stencil & s ) const [inline]`

Definition at line 134 of file Gauss1DsModel.h.

References `mappel::Gauss1DsModel::Stencil::bg()`, `mappel::Gauss1DsModel::Stencil::l()`, and `mappel::Gauss1DsModel::Stencil::X`.

9.11.4.64 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const [inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

9.11.4.65 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const [inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

9.11.4.66 `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name ) [inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.11.4.67 `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const [inherited]`

Definition at line 59 of file MCMCAdaptor1Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1Ds::eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.11.4.68 `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const [inherited]`

Definition at line 77 of file MCMCAdaptor1Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1Ds::eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.11.4.69 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.11.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline]`, `[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.11.4.71 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` `[inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.11.4.72 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.11.4.73 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline]`,  
`[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.11.4.74 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline]`,  
`[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.



9.11.4.75 `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams )` `[inline]`, `[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

9.11.4.76 `template<class ImT > void ImageFormat1DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const` `[inherited]`

Definition at line 115 of file `ImageFormat1DBase.h`.

9.11.4.77 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 )` `[inherited]`

Definition at line 65 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.11.4.78 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )` `[inherited]`

Definition at line 226 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `set_min_sigma()`.

9.11.4.79 `void mappel::Gauss1DsModel::set_max_sigma ( double max_sigma )`

Definition at line 135 of file `Gauss1DsModel.cpp`.

References `mappel::PointEmitterModel::check_psf_sigma()`, `get_min_sigma()`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::set_ubound()`.

Referenced by `set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::Gauss2DsModel::set_min_sigma()`.

9.11.4.80 `void mappel::Gauss1DsModel::set_max_sigma ( const VecT & max_sigma )`

Definition at line 153 of file `Gauss1DsModel.cpp`.

References `set_max_sigma()`.

9.11.4.81 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` [protected],[inherited]

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.11.4.82 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` [inherited]

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.11.4.83 `void mappel::Gauss1DsModel::set_min_sigma ( double min_sigma )`

Definition at line 122 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `get_max_sigma()`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::set_lbound()`.

Referenced by `set_min_sigma()`, and `mappel::Gauss2DsModel::set_min_sigma()`.

9.11.4.84 `void mappel::Gauss1DsModel::set_min_sigma ( const VecT & min_sigma )`

Definition at line 148 of file Gauss1DsModel.cpp.

References `set_min_sigma()`.

9.11.4.85 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.11.4.86 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )` [inherited]

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

9.11.4.87 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ )` [inherited]

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.11.4.88 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.11.4.89 `void ImageFormat1DBase::set_size ( const ImageSizeT & size_ ) [inherited]`

Definition at line 30 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::check\_size(), and mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::set\_size(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DsModel::set\_size().

9.11.4.90 `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz ) [inline],[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::set\_size().

9.11.4.91 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by set\_max\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.11.4.92 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::make\_stencil(), make\_stencil(), mappel::Gauss2DModel::make\_stencil(), mappel::Gauss2DsxModel::make\_stencil(), mappel::Gauss2DsModel::make\_stencil(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.11.4.93 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const [inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::theta\_in\_bounds().

### 9.11.5 Member Data Documentation

**9.11.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2D↔DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

**9.11.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file `PointEmitterModel.h`.

**9.11.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file `PointEmitterModel.h`.

**9.11.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file `PointEmitterModel.h`.

**9.11.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.11.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file `PointEmitterModel.h`.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.11.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.11.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

9.11.5.9 `const std::string mappel::Gauss1DsModel::DefaultPriorType = "Normal"` `[static]`

Definition at line 51 of file Gauss1DsModel.h.

Referenced by `operator=()`.

9.11.5.10 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.11.5.11 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.11.5.12 `double mappel::MCMCAdaptor1D::eta_I = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_I` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

#### 9.11.5.13 `double mappel::MCMCAdaptor1Ds::eta_sigma = 1` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor1Ds.h`.

Referenced by `mappel::MCMCAdaptor1Ds::get_stats()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor1Ds::operator=()`, and `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`.

#### 9.11.5.14 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

#### 9.11.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

#### 9.11.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

#### 9.11.5.17 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

#### 9.11.5.18 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.11.5.19** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.11.5.20** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.11.5.21** `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.11.5.22** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static], [inherited]`

Number of image dimensions.

Definition at line 38 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

**9.11.5.23** `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

**9.11.5.24** `IdxT mappel::PointEmitterModel::num_params` `[protected], [inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.11.5.25 `IdxT mappel::MCMCAdaptorBase::num_phases` [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

#### 9.11.5.26 `CompositeDist mappel::PointEmitterModel::prior` [protected],[inherited]

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `get_max_sigma()`, `get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `set_max_sigma()`, `set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.11.5.27 `const StringVecT mappel::Gauss1DsModel::prior_types` [static]

**Initial value:**

```
= { "Beta",
    "Normal"
}
```

Definition at line 50 of file `Gauss1DsModel.h`.

Referenced by `operator=()`.

#### 9.11.5.28 `double mappel::MCMCAdaptorBase::sigma_scale` [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.



#### 9.11.5.29 ImageSizeT mappel::ImageFormat1DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::check\_image\_shape(), mappel::Gauss1DsModel::Stencil::compute\_derivatives(), mappel::Gauss1DModel::Stencil::compute\_derivatives(), mappel::ImageFormat1DBase::get\_num\_pixels(), mappel::ImageFormat1DBase::get\_size(), mappel::Gauss1DModel::initial\_theta\_estimate(), initial\_theta\_estimate(), mappel::ImageFormat1DBase::make\_image(), mappel::ImageFormat1DBase::make\_image\_stack(), mappel::ImageFormat1DBase::set\_size(), mappel::Gauss1DsModel::Stencil::Stencil(), and mappel::Gauss1DModel::Stencil::Stencil().

#### 9.11.5.30 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

The documentation for this class was generated from the following files:

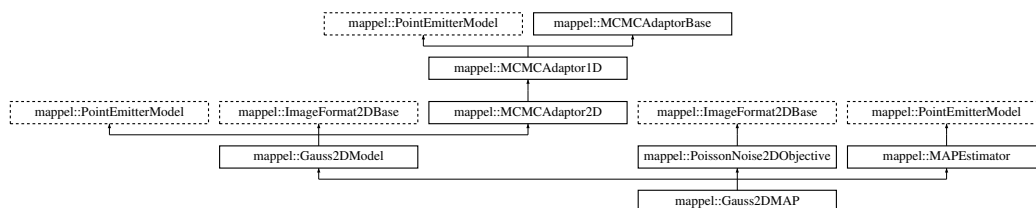
- [Gauss1DsModel.h](#)
- [Gauss1DsModel.cpp](#)

## 9.12 mappel::Gauss2DMP Class Reference

A 2D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DMP.h>
```

Inheritance diagram for mappel::Gauss2DMP:



## Public Types

- using [Gauss1DSumModelT](#) = [Gauss1DMAP](#)
- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >
- using [ModelDataT](#) = [ImageT](#)
- using [ModelDataStackT](#) = [ImageStackT](#)

## Public Member Functions

- [Gauss2DMAP](#) ([ImageCoordT](#) size, double [psf\\_sigma](#), const std::string &prior\_type=[DefaultPriorType](#))
- [Gauss2DMAP](#) (const [ImageSizeT](#) &size, double [psf\\_sigma](#), const std::string &prior\_type=[DefaultPriorType](#))
- template<class IntType , class FloatType >  
[Gauss2DMAP](#) (const arma::Col< IntType > &size, const arma::Col< FloatType > &[psf\\_sigma](#), const std::string &prior\_type=[DefaultPriorType](#))
- [Gauss2DMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &[psf\\_sigma](#), CompositeDist &&prior)
- [Gauss2DMAP](#) ([ImageSizeT](#) &&size, [VecT](#) &&[psf\\_sigma](#), CompositeDist &&prior)
- [Gauss2DMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &[psf\\_sigma](#), const CompositeDist &prior)
- [Gauss2DMAP](#) (const [Gauss2DMAP](#) &o)
- [Gauss2DMAP](#) & operator= (const [Gauss2DMAP](#) &o)
- [Gauss2DMAP](#) ([Gauss2DMAP](#) &&o)
- [Gauss2DMAP](#) & operator= ([Gauss2DMAP](#) &&o)
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- const [VecT](#) & [get\\_psf\\_sigma](#) () const
- double [get\\_psf\\_sigma](#) (IdxT idx) const
- void [set\\_psf\\_sigma](#) (double new\_psf\_sigma)
- void [set\\_psf\\_sigma](#) (const [VecT](#) &new\_psf\_sigma)
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &theta, bool compute\_derivatives=true) const  
*Make a new Model::Stencil object at theta.*
- double [pixel\\_model\\_value](#) (int i, int j, const [Stencil](#) &s) const
- void [pixel\\_grad](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad) const

- void `pixel_grad2` (int i, int j, const `Stencil` &s, `ParamT` &pgrad2) const
- void `pixel_hess` (int i, int j, const `Stencil` &s, `MatT` &hess) const
- void `pixel_hess_update` (int i, int j, const `Stencil` &s, double dm\_ratio\_m1, double dmm\_ratio, `ParamT` &grad, `MatT` &hess) const
- *pixel derivative inner loop calculations.*
- `Stencil initial_theta_estimate` (const `ImageT` &im) const
- *Fast, heuristic estimate of initial theta.*
- `Stencil initial_theta_estimate` (const `ImageT` &im, const `ParamT` &theta\_init) const
- `Stencil initial_theta_estimate` (const `ImageT` &im, const `ParamT` &theta\_init, const std::string &estimator) const
- `Gauss1DSumModelT debug_internal_sum_model_x` () const
- `Gauss1DSumModelT debug_internal_sum_model_y` () const
- `IdxT get_num_params` () const
- void `check_param_shape` (const `ParamT` &theta) const
- void `check_param_shape` (const `ParamVecT` &theta) const
- void `check_psf_sigma` (double `psf_sigma`) const
- void `check_psf_sigma` (const `VecT` &`psf_sigma`) const
- `ParamT make_param` () const
- template<class FillT >  
  `ParamT make_param` (FillT fill) const
- `ParamVecT make_param_stack` (`IdxT` n) const
- template<class FillT >  
  `ParamVecT make_param_stack` (`IdxT` n, FillT fill) const
- `MatT make_param_mat` () const
- template<class FillT >  
  `MatT make_param_mat` (FillT fill) const
- `CubeT make_param_mat_stack` (`IdxT` n) const
- template<class FillT >  
  `CubeT make_param_mat_stack` (`IdxT` n, FillT fill) const
- `CompositeDist & get_prior` ()
- const `CompositeDist & get_prior` () const
- `IdxT get_num_hyperparams` () const
- `VecT get_hyperparams` () const
- bool `has_hyperparam` (const std::string &name) const
- double `get_hyperparam_value` (const std::string &name) const
- int `get_hyperparam_index` (const std::string &name) const
- void `set_hyperparam_value` (const std::string &name, double value)
- void `rename_hyperparam` (const std::string &old\_name, const std::string &new\_name)
- `StringVecT get_param_names` () const
- void `set_param_names` (const `StringVecT` &desc)
- `StringVecT get_hyperparam_names` () const
- void `set_hyperparam_names` (const `StringVecT` &desc)
- template<class RngT >  
  `ParamT sample_prior` (RngT &rng) const
- `ParamT sample_prior` () const
- void `set_bounds` (const `ParamT` &lbound, const `ParamT` &ubound)
- void `set_lbound` (const `ParamT` &lbound)
- void `set_ubound` (const `ParamT` &ubound)
- const `ParamT & get_lbound` () const
- const `ParamT & get_ubound` () const
- bool `theta_in_bounds` (const `ParamT` &theta) const
- void `bound_theta` (`ParamT` &theta, double epsilon=`bounds_epsilon`) const

- [ParamT bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT make\\_image](#) () const
- [ImageStackT make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- `template<class ImT >`  
`void set\_image\_in\_stack (ImageStackT &stack, ImageCoordT n, const ImT &im) const`
- `const ImageSizeT & get\_size () const`
- `ImageCoordT get\_size (IdxT idx) const`
- `ImageCoordT get\_num\_pixels () const`
- `void check\_image\_shape (const ImageT &im) const`  
*Check the shape of a single images is correct for model size.*
- `void check\_image\_shape (const ImageStackT &ims) const`  
*Check the shape of a stack of images is correct for model size.*
- `void sample\_mcmc\_candidate (IdxT sample_index, ParamT &candidate, double step_scale=1.0) const`
- `void sample\_mcmc\_candidate (IdxT sample_index, ParamT &candidate, const IdxVecT &fixed_parameters_↵  
mask, double step_scale=1.0) const`
- `void set\_intensity\_mcmc\_sampling (double eta_l=-1)`
- `void set\_background\_mcmc\_sampling (double eta_bg=-1)`
- `void set\_mcmc\_sigma\_scale (double scale)`
- `double get\_mcmc\_sigma\_scale () const`
- `IdxT get\_mcmc\_num\_phases () const`

### Static Public Member Functions

- static [CompositeDist](#) [make\\_default\\_prior](#) (const [ImageSizeT](#) &size, const std::string &prior\_type)
- static [CompositeDist](#) [make\\_default\\_prior\\_beta\\_position](#) (const [ImageSizeT](#) &size)
- static [CompositeDist](#) [make\\_default\\_prior\\_normal\\_position](#) (const [ImageSizeT](#) &size)
- static [CompositeDist](#) [make\\_prior\\_beta\\_position](#) (const [ImageSizeT](#) &size, double beta\_xpos, double beta\_ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static [CompositeDist](#) [make\\_prior\\_normal\\_position](#) (const [ImageSizeT](#) &size, double sigma\_xpos, double beta\_↵  
ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static [prior\\_hessian::TruncatedNormalDist](#) [make\\_prior\\_component\\_position\\_normal](#) ([IdxT](#) size, double pos\_↵  
sigma=[default\\_sigma\\_pos](#))
- static [prior\\_hessian::ScaledSymmetricBetaDist](#) [make\\_prior\\_component\\_position\\_beta](#) ([IdxT](#) size, double pos\_↵  
beta=[default\\_beta\\_pos](#))
- static [prior\\_hessian::TruncatedGammaDist](#) [make\\_prior\\_component\\_intensity](#) (double mean=[default\\_mean\\_↵  
l](#), double kappa=[default\\_intensity\\_kappa](#))
- static [prior\\_hessian::TruncatedParetoDist](#) [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵  
sigma, double alpha=[default\\_alpha\\_sigma](#))
- static void [set\\_rng\\_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)

*Check the size argument for the model.*

## Static Public Attributes

- static const std::string `name`
- static const `StringVecT` `prior_types`
- static const std::string `DefaultPriorType` = "Normal"
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1
- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2
- static const `ImageCoordT` `num_dim` =2
- static const `ImageCoordT` `global_min_size` =3
- static const `ImageCoordT` `global_max_size` =512
- static const double `global_default_mcmc_sigma_scale` = 0.05
- static const double `global_max_mcmc_sigma_scale` = 0.5
- static const std::vector< std::string > `estimator_names`

## Protected Member Functions

- void `update_internal_1Dsum_estimators` ()
- void `set_mcmc_num_phases` (`IdxT` num\_phases)

## Static Protected Member Functions

- static `Gauss1DSumModelT` `make_internal_1Dsum_estimator` (`IdxT` dim, const `ImageSizeT` &size, const `VecT` &psf\_sigma, const `CompositeDist` &prior)

## Protected Attributes

- `VecT` `psf_sigma`
- `Gauss1DSumModelT` `x_model`
- `Gauss1DSumModelT` `y_model`
- `CompositeDist` `prior`
- `IdxT` `num_params`
- `IdxT` `num_hyperparams`
- `ParamT` `lbound`
- `ParamT` `ubound`
- `ImageSizeT` `size`
- double `eta_y` =0
- double `eta_x` =0
- double `eta_l` =0
- double `eta_bg` =0
- `IdxT` `num_phases`
- double `sigma_scale`

### 9.12.1 Detailed Description

A 2D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.

Model: [Gauss2DModel](#) a 2D gaussian PSF with fixed psf\_sigma Objective: [PoissonNoise2DObjective](#) - Poisson noise model for 2D Estimator: [MAPEstimator](#) - Maximum a-posteriori estimator

Definition at line 23 of file Gauss2DMAP.h.

### 9.12.2 Member Typedef Documentation

**9.12.2.1** `using mappel::Gauss2DModel::Gauss1DSumModelT = Gauss1DMAP` `[inherited]`

Definition at line 23 of file Gauss2DModel.h.

**9.12.2.2** `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` `[inherited]`

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

**9.12.2.3** `using mappel::ImageFormat2DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

**9.12.2.4** `template<class PixelT > using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>`  
`[inherited]`

Shape of the data type for a single image

Definition at line 32 of file ImageFormat2DBase.h.

**9.12.2.5** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>`  
`[inherited]`

Shape of the data type to store a single image's coordinates

Definition at line 27 of file ImageFormat2DBase.h.

**9.12.2.6** `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`  
`[inherited]`

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

9.12.2.7 `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>`  
`[inherited]`

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

9.12.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

9.12.2.9 `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

9.12.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

9.12.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

9.12.2.12 `using mappel::PoissonNoise2DObjective::ModelDataStackT = ImageStackT` `[inherited]`

Objective function data stack type: 2D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise2DObjective.h.

9.12.2.13 `using mappel::PoissonNoise2DObjective::ModelDataT = ImageT` `[inherited]`

Objective function data type: 2D double precision image, gain-corrected to approximate photons counts

Definition at line 24 of file PoissonNoise2DObjective.h.

9.12.2.14 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.12.2.15 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

9.12.2.16 `using mappel::Gauss2DModel::StencilVecT = std::vector<Stencil>` [inherited]

Definition at line 47 of file Gauss2DModel.h.

### 9.12.3 Constructor & Destructor Documentation

9.12.3.1 `mappel::Gauss2DMap::Gauss2DMap ( ImageCoordT size, double psf_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 11 of file Gauss2DMap.cpp.

9.12.3.2 `mappel::Gauss2DMap::Gauss2DMap ( const ImageSizeT & size, double psf_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 15 of file Gauss2DMap.cpp.

9.12.3.3 `template<class IntType, class FloatType> mappel::Gauss2DMap::Gauss2DMap ( const arma::Col< IntType > & size, const arma::Col< FloatType > & psf_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 41 of file Gauss2DMap.h.

9.12.3.4 `mappel::Gauss2DMap::Gauss2DMap ( const ImageSizeT & size, const VecT & psf_sigma, CompositeDist && prior )`

Definition at line 27 of file Gauss2DMap.cpp.

9.12.3.5 `mappel::Gauss2DMap::Gauss2DMap ( ImageSizeT && size, VecT && psf_sigma, CompositeDist && prior )`

Definition at line 19 of file Gauss2DMap.cpp.

9.12.3.6 `mappel::Gauss2DMap::Gauss2DMap ( const ImageSizeT & size, const VecT & psf_sigma, const CompositeDist & prior )`

Definition at line 35 of file Gauss2DMap.cpp.



## 9.12.3.7 mappel::Gauss2DMap::Gauss2DMap ( const Gauss2DMap &amp; o )

Definition at line 43 of file Gauss2DMap.cpp.

## 9.12.3.8 mappel::Gauss2DMap::Gauss2DMap ( Gauss2DMap &amp;&amp; o )

Definition at line 51 of file Gauss2DMap.cpp.

## 9.12.4 Member Function Documentation

## 9.12.4.1 void mappel::PointEmitterModel::bound\_theta ( ParamT &amp; theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

## 9.12.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT &amp; theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

## 9.12.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT &amp; theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

## 9.12.4.4 void mappel::ImageFormat2DBase::check\_image\_shape ( const ImageT &amp; im ) const [inherited]

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

## 9.12.4.5 void mappel::ImageFormat2DBase::check\_image\_shape ( const ImageStackT &amp; ims ) const [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.12.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

**9.12.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

**9.12.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

**9.12.4.9** `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

**9.12.4.10** `void mappel::ImageFormat2DBase::check_size ( const ImageSizeT & size )` [static],[inherited]

Check the size argument for the model.

Definition at line 60 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::global\_max\_size, and mappel::ImageFormat2DBase::global\_min\_size.

Referenced by mappel::ImageFormat2DBase::ImageFormat2DBase(), and mappel::ImageFormat2DBase::set\_size().

**9.12.4.11** `Gauss1DSumModelT mappel::Gauss2DModel::debug_internal_sum_model_x ( ) const` [inline], [inherited]

Definition at line 89 of file Gauss2DModel.h.

References mappel::Gauss2DModel::x\_model.

**9.12.4.12** `Gauss1DSumModelT` `mappel::Gauss2DModel::debug_internal_sum_model_y ( ) const` `[inline]`,  
`[inherited]`

Definition at line 90 of file `Gauss2DModel.h`.

References `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DModel::operator=()`, `mappel::PointEmitterModel::prior`, `mappel::Gauss2DModel::psf_sigma`, `mappel::ImageFormat2DBase::size`, `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DModel::y_model`.

**9.12.4.13** `int` `mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 236 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.12.4.14** `StringVecT` `mappel::PointEmitterModel::get_hyperparam_names ( ) const` `[inline]`, `[inherited]`

Definition at line 256 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.12.4.15** `double` `mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 232 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.12.4.16** `PointEmitterModel::ParamT` `mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.12.4.17** `ImageFormat2DBase::ImageT` `mappel::ImageFormat2DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` `[inline]`, `[inherited]`

Definition at line 106 of file `ImageFormat2DBase.h`.

**9.12.4.18** `const PointEmitterModel::ParamT &` `mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.12.4.19** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

**9.12.4.20** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

**9.12.4.21** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

**9.12.4.22** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.12.4.23** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_num_pixels ( ) const` `[inline],[inherited]`

Definition at line 79 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

**9.12.4.24** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.12.4.25** `CompositeDist & mappel::PointEmitterModel::get_prior ( )` `[inline],[inherited]`

Definition at line 200 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

9.12.4.26 `const CompositeDist & mappel::PointEmitterModel::get_prior ( ) const` `[inline],[inherited]`

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.12.4.27 `const VecT & mappel::Gauss2DModel::get_psf_sigma ( ) const` `[inline],[inherited]`

Definition at line 142 of file Gauss2DModel.h.

References mappel::Gauss2DModel::psf\_sigma.

Referenced by mappel::Gauss2DModel::get\_stats().

9.12.4.28 `double mappel::Gauss2DModel::get_psf_sigma ( IdxT idx ) const` `[inherited]`

Definition at line 132 of file Gauss2DModel.cpp.

References mappel::Gauss2DModel::DefaultPriorType, mappel::Gauss2DModel::prior\_types, and mappel::Gauss2DModel::psf\_sigma.

9.12.4.29 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ( )` `[static],[inherited]`

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.12.4.30 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ( )` `[static],[inherited]`

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.12.4.31 `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size ( ) const` `[inline],[inherited]`

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

9.12.4.32 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size ( IdxT idx ) const` `[inherited]`

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.12.4.33 ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::get\_size\_image\_stack ( const ImageStackT & stack ) const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

**9.12.4.34 StatsT** mappel::Gauss2DModel::get\_stats ( ) const [inherited]

Definition at line 268 of file Gauss2DModel.cpp.

References mappel::Gauss2DModel::get\_psf\_sigma(), mappel::MCMCAdaptor2D::get\_stats(), mappel::ImageFormat2DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

**9.12.4.35 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const** [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

**9.12.4.36 bool** mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.12.4.37 Gauss2DModel::Stencil** mappel::Gauss2DModel::initial\_theta\_estimate ( const ImageT & im ) const [inline],[inherited]

Fast, heuristic estimate of initial theta.

Definition at line 194 of file Gauss2DModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::PointEmitterModel::make\_param().

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate().

**9.12.4.38 Gauss2DModel::Stencil** mappel::Gauss2DModel::initial\_theta\_estimate ( const ImageT & im, const ParamT & theta\_init ) const [inline],[inherited]

Definition at line 201 of file Gauss2DModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::Gauss2DModel::initial\_theta\_estimate().

**9.12.4.39** `Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init, const std::string & estimator ) const` `[inherited]`

Definition at line 303 of file Gauss2DModel.cpp.

References `mappel::Gauss2DModel::Stencil::bg()`, `mappel::methods::estimate_max()`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::PointEmitterModel::lbound`, `mappel::Gauss2DModel::make_stencil()`, `mappel::PointEmitterModel::num_params`, `mappel::ImageFormat2DBase::size`, `mappel::estimator::MLEData::theta`, `mappel::PointEmitterModel::theta_in_bounds()`, `mappel::PointEmitterModel::ubound`, `mappel::Gauss2DModel::x_model`, and `mappel::Gauss2DModel::y_model`.

**9.12.4.40** `CompositeDist mappel::Gauss2DModel::make_default_prior ( const ImageSizeT & size, const std::string & prior_type )` `[static]`, `[inherited]`

Definition at line 150 of file Gauss2DModel.cpp.

References `mappel::istarts_with()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, and `mappel::Gauss2DModel::make_default_prior_normal_position()`.

**9.12.4.41** `CompositeDist mappel::Gauss2DModel::make_default_prior_beta_position ( const ImageSizeT & size )` `[static]`, `[inherited]`

Definition at line 171 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::Gauss2DModel::make_default_prior()`.

**9.12.4.42** `CompositeDist mappel::Gauss2DModel::make_default_prior_normal_position ( const ImageSizeT & size )` `[static]`, `[inherited]`

Definition at line 182 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::Gauss2DModel::make_default_prior()`.

**9.12.4.43** `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image ( ) const` `[inline]`, `[inherited]`

Definition at line 85 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

**9.12.4.44 ImageFormat2DBase::ImageStackT** mappel::ImageFormat2DBase::make\_image\_stack ( ImageCoordT *n* ) const  
[inline],[inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.12.4.45 Gauss2DModel::Gauss1DSumModelT** mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator ( IdxT *dim*,  
const ImageSizeT & *size*, const VecT & *psf\_sigma*, const CompositeDist & *prior* ) [static],[protected],  
[inherited]

Definition at line 62 of file Gauss2DModel.cpp.

References mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_normal\_↵  
position(), mappel::Gauss2DModel::psf\_sigma, and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(),  
and mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators().

**9.12.4.46 PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline],[inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(),  
mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::↵  
Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel↵  
::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_↵  
update().

**9.12.4.47 template<class FillT > PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( FillT *fill* ) const  
[inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.12.4.48 MatT** mappel::PointEmitterModel::make\_param\_mat ( ) const [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.12.4.49 template<class FillT > MatT** mappel::PointEmitterModel::make\_param\_mat ( FillT *fill* ) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.



9.12.4.50 **CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n* ) const [inline], [inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.12.4.51 **template<class FillT > CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n*, FillT *fill* ) const [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.12.4.52 **PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make\_param\_stack ( IdxT *n* ) const [inline], [inherited]

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.12.4.53 **template<class FillT > PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make\_param\_stack ( IdxT *n*, FillT *fill* ) const [inherited]

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.12.4.54 **CompositeDist** mappel::Gauss2DModel::make\_prior\_beta\_position ( const ImageSizeT & *size*, double *beta\_xpos*, double *beta\_ypos*, double *mean\_l*, double *kappa\_l*, double *mean\_bg*, double *kappa\_bg* ) [static], [inherited]

Definition at line 193 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), and mappel::ImageFormat2DBase::size.

9.12.4.55 **prior\_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make\_prior\_component\_intensity ( double *mean* = default\_mean\_l, double *kappa* = default\_intensity\_kappa ) [static], [inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_l.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.12.4.56 **prior\_hessian::ScaledSymmetricBetaDist** **mappel::PointEmitterModel::make\_prior\_component\_position\_beta** ( *IdxT size*, *double pos\_beta = default\_beta\_pos* ) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by **mappel::Gauss1DsModel::make\_default\_prior\_beta\_position()**, **mappel::Gauss2DModel::make\_default\_prior\_beta\_position()**, **mappel::Gauss1DModel::make\_default\_prior\_beta\_position()**, **mappel::Gauss2DsModel::make\_default\_prior\_beta\_position()**, **mappel::Gauss1DsModel::make\_prior\_beta\_position()**, **mappel::Gauss2DModel::make\_prior\_beta\_position()**, **mappel::Gauss1DModel::make\_prior\_beta\_position()**, and **mappel::Gauss2DsModel::make\_prior\_beta\_position()**.

9.12.4.57 **prior\_hessian::TruncatedNormalDist** **mappel::PointEmitterModel::make\_prior\_component\_position\_normal** ( *IdxT size*, *double pos\_sigma = default\_sigma\_pos* ) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by **mappel::Gauss1DsModel::make\_default\_prior\_normal\_position()**, **mappel::Gauss2DModel::make\_default\_prior\_normal\_position()**, **mappel::Gauss1DModel::make\_default\_prior\_normal\_position()**, **mappel::Gauss2DsModel::make\_default\_prior\_normal\_position()**, **mappel::Gauss1DsModel::make\_prior\_normal\_position()**, **mappel::Gauss2DModel::make\_prior\_normal\_position()**, **mappel::Gauss1DsModel::make\_prior\_normal\_position()**, and **mappel::Gauss2DsModel::make\_prior\_normal\_position()**.

9.12.4.58 **prior\_hessian::TruncatedParetoDist** **mappel::PointEmitterModel::make\_prior\_component\_sigma** ( *double min\_sigma*, *double max\_sigma*, *double alpha = default\_alpha\_sigma* ) [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by **mappel::Gauss1DsModel::make\_default\_prior\_beta\_position()**, **mappel::Gauss2DsModel::make\_default\_prior\_beta\_position()**, **mappel::Gauss1DsModel::make\_default\_prior\_normal\_position()**, **mappel::Gauss2DsModel::make\_default\_prior\_normal\_position()**, **mappel::Gauss1DsModel::make\_prior\_beta\_position()**, **mappel::Gauss2DsModel::make\_prior\_beta\_position()**, **mappel::Gauss1DsModel::make\_prior\_normal\_position()**, and **mappel::Gauss2DsModel::make\_prior\_normal\_position()**.

9.12.4.59 **CompositeDist** **mappel::Gauss2DModel::make\_prior\_normal\_position** ( *const ImageSizeT & size*, *double sigma\_xpos*, *double beta\_ypos*, *double mean\_l*, *double kappa\_l*, *double mean\_bg*, *double kappa\_bg* ) [static],[inherited]

Definition at line 206 of file Gauss2DModel.cpp.

References **mappel::PointEmitterModel::make\_prior\_component\_intensity()**, **mappel::PointEmitterModel::make\_prior\_component\_position\_normal()**, and **mappel::ImageFormat2DBase::size**.

9.12.4.60 **Gauss2DModel::Stencil** **mappel::Gauss2DModel::make\_stencil** ( *const ParamT & theta*, *bool compute\_derivatives = true* ) **const** [inline],[inherited]

Make a new **Model::Stencil** object at *theta*.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular *theta* (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a **const Stencil** reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not **model.theta\_in\_bounds(theta)**.

If derivatives will not be computed with this stencil set **compute\_derivatives=false**

## Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

## Returns

A new [Stencil](#) object ready to compute with

Definition at line 131 of file Gauss2DModel.h.

References [mappel::Gauss2DModel::Stencil::Stencil\(\)](#), and [mappel::PointEmitterModel::theta\\_in\\_bounds\(\)](#).

Referenced by [mappel::Gauss2DModel::initial\\_theta\\_estimate\(\)](#).

#### 9.12.4.61 Gauss2DModel & mappel::Gauss2DModel::operator= ( const Gauss2DModel & o )

Definition at line 59 of file Gauss2DModel.cpp.

References [mappel::MAPEstimator::operator=\(\)](#), [mappel::PoissonNoise2DObjective::operator=\(\)](#), [mappel::ImageFormat2DBase::operator=\(\)](#), [mappel::Gauss2DModel::operator=\(\)](#), and [mappel::PointEmitterModel::operator=\(\)](#).

#### 9.12.4.62 Gauss2DModel & mappel::Gauss2DModel::operator= ( Gauss2DModel && o )

Definition at line 70 of file Gauss2DModel.cpp.

References [mappel::MAPEstimator::operator=\(\)](#), [mappel::PoissonNoise2DObjective::operator=\(\)](#), [mappel::ImageFormat2DBase::operator=\(\)](#), [mappel::Gauss2DModel::operator=\(\)](#), and [mappel::PointEmitterModel::operator=\(\)](#).

#### 9.12.4.63 void mappel::Gauss2DModel::pixel\_grad ( int i, int j, const Stencil & s, ParamT & pgrad ) const [inline], [inherited]

Definition at line 159 of file Gauss2DModel.h.

References [mappel::Gauss2DModel::Stencil::DX](#), [mappel::Gauss2DModel::Stencil::DY](#), [mappel::Gauss2DModel::Stencil::I\(\)](#), [mappel::Gauss2DModel::Stencil::X](#), and [mappel::Gauss2DModel::Stencil::Y](#).

Referenced by [mappel::Gauss2DModel::pixel\\_hess\\_update\(\)](#).

#### 9.12.4.64 void mappel::Gauss2DModel::pixel\_grad2 ( int i, int j, const Stencil & s, ParamT & pgrad2 ) const [inline], [inherited]

Definition at line 170 of file Gauss2DModel.h.

References [mappel::Gauss2DModel::Stencil::DXS](#), [mappel::Gauss2DModel::Stencil::DYS](#), [mappel::Gauss2DModel::Stencil::I\(\)](#), [mappel::Gauss2DModel::psf\\_sigma](#), [mappel::Gauss2DModel::Stencil::X](#), and [mappel::Gauss2DModel::Stencil::Y](#).

**9.12.4.65** `void mappel::Gauss2DModel::pixel_hess ( int i, int j, const Stencil & s, MatT & hess ) const` `[inline]`,  
`[inherited]`

Definition at line 181 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

**9.12.4.66** `void mappel::Gauss2DModel::pixel_hess_update ( int i, int j, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess ) const` `[inherited]`

pixel derivative inner loop calculations.

Definition at line 282 of file Gauss2DModel.cpp.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::PointEmitterModel::make_param()`, `mappel::Gauss2DModel::pixel_grad()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

**9.12.4.67** `double mappel::Gauss2DModel::pixel_model_value ( int i, int j, const Stencil & s ) const` `[inline]`,  
`[inherited]`

Definition at line 152 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::bg()`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

**9.12.4.68** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const` `[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.12.4.69** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.12.4.70** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )` `[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.12.4.71 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const` [inherited]

Definition at line 59 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_`  
`x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.12.4.72 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const` [inherited]

Definition at line 74 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_`  
`x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.12.4.73 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const` [inherited]

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.12.4.74 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` [inline],[inherited]

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.12.4.75 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::Point`  
`EmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_`  
`ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.12.4.76 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitter`  
`Model::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.12.4.77 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc ) [inline], [inherited]`

Definition at line 260 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.12.4.78 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.12.4.79 `void mappel::Gauss2DModel::set_hyperparams ( const VecT & hyperparams ) [inherited]`

Definition at line 109 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::set\_hyperparams(), and mappel::Gauss2DModel::update\_internal\_1Dsum\_↵ estimators().

9.12.4.80 `template<class ImT > void mappel::ImageFormat2DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]`

Definition at line 113 of file ImageFormat2DBase.h.

9.12.4.81 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 ) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_mean\_I, mappel::MCMCAdaptor1D::eta\_I, mappel::PointEmitter↵ Model::get\_hyperparam\_value(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

9.12.4.82 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitter↵ Model::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.12.4.83 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases ) [protected], [inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

Referenced by mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), and mappel::MCMCAdaptor2Ds::MCMCAdaptor2↵ Ds().

9.12.4.84 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` `[inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_↵_scale`.

9.12.4.85 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` `[inline], [inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.12.4.86 `void mappel::Gauss2DModel::set_prior ( CompositeDist && prior_ )` `[inherited]`

Definition at line 97 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `mappel::Gauss2DModel::update_internal_1Dsum_↵_estimators()`.

9.12.4.87 `void mappel::Gauss2DModel::set_prior ( const CompositeDist & prior_ )` `[inherited]`

Definition at line 103 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `mappel::Gauss2DModel::update_internal_1Dsum_↵_estimators()`.

9.12.4.88 `void mappel::Gauss2DModel::set_psf_sigma ( double new_psf_sigma )` `[inline], [inherited]`

Definition at line 146 of file Gauss2DModel.h.

9.12.4.89 `void mappel::Gauss2DModel::set_psf_sigma ( const VecT & new_psf_sigma )` `[inherited]`

Definition at line 123 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::Gauss1D↵_Model::set_psf_sigma()`, `mappel::Gauss2DModel::x_model`, and `mappel::Gauss2DModel::y_model`.

9.12.4.90 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` `[static], [inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.12.4.91 `void mappel::Gauss2DModel::set_size ( const ImageSizeT & size_ )` `[inherited]`

Definition at line 115 of file Gauss2DModel.cpp.

References `mappel::ImageFormat2DBase::set_size()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Image↵_Format2DBase::size`, `mappel::Gauss2DModel::x_model`, and `mappel::Gauss2DModel::y_model`.

**9.12.4.92** `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

**9.12.4.93** `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const` [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.12.4.94** `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const` [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.12.4.95** `void mappel::Gauss2DModel::update_internal_1Dsum_estimators ( )` [protected],[inherited]

Definition at line 91 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::get_prior()`, `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::ImageFormat2DBase::size`, `mappel::Gauss2DModel::x_model`, and `mappel::Gauss2DModel::y_model`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DModel::set_prior()`.

## 9.12.5 Member Data Documentation

**9.12.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static],[inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.



9.12.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.12.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.12.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

9.12.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.12.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` [static],[inherited]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.12.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` [static],[inherited]

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.12.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.12.5.9** `const std::string mappel::Gauss2DModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 51 of file Gauss2DModel.h.

Referenced by `mappel::Gauss2DModel::get_psf_sigma()`.

**9.12.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.12.5.11** `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` `[static], [inherited]`

Definition at line 23 of file PoissonNoise2DObjective.h.

**9.12.5.12** `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.12.5.13** `double mappel::MCMCAdaptor1D::eta_I = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_I` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.12.5.14 `double mappel::MCMCAdaptor1D::eta_x=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

9.12.5.15 `double mappel::MCMCAdaptor2D::eta_y=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor2D.h`.

Referenced by `mappel::MCMCAdaptor2D::get_stats()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2D::operator=()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`.

9.12.5.16 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static]`, `[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.12.5.17 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static]`, `[inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

9.12.5.18 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static]`, `[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.12.5.19 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size=512` `[static]`, `[inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file `ImageFormat2DBase.h`.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.12.5.20** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.12.5.21** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file `ImageFormat2DBase.h`.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.12.5.22** `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.12.5.23** `const std::string mappel::Gauss2DMap::name` `[static]`

Definition at line 37 of file `Gauss2DMap.h`.

**9.12.5.24** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim = 2` `[static], [inherited]`

Number of image dimensions.

Definition at line 37 of file `ImageFormat2DBase.h`.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

**9.12.5.25** `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

#### 9.12.5.26 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.12.5.27 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

#### 9.12.5.28 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.12.5.29 `const StringVecT mappel::Gauss2DModel::prior_types` `[static]`, `[inherited]`

**Initial value:**

```
= { "Beta",
    "Normal"
}
```

Definition at line 50 of file `Gauss2DModel.h`.

Referenced by `mappel::Gauss2DModel::get_psf_sigma()`.

### 9.12.5.30 VecT mappel::Gauss2DModel::psf\_sigma [protected],[inherited]

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 104 of file Gauss2DModel.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_←  
\_heuristic\_compute\_estimate(), mappel::Gauss2DModel::Stencil::compute\_derivatives(), mappel::Gauss2DModel←  
::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DModel::get\_psf\_←  
sigma(), mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DModel::operator=(), mappel←  
::Gauss2DModel::pixel\_grad2(), mappel::Gauss2DModel::pixel\_hess(), mappel::Gauss2DModel::pixel\_hess\_update(),  
mappel::Gauss2DModel::set\_psf\_sigma(), mappel::Gauss2DModel::Stencil::Stencil(), and mappel::Gauss2DModel←  
::update\_internal\_1Dsum\_estimators().

### 9.12.5.31 double mappel::MCMCAdaptorBase::sigma\_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(),  
mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMC←  
Adaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set←  
\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMC←  
AdaptorBase::set\_mcmc\_sigma\_scale().

### 9.12.5.32 ImageSizeT mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_←  
\_heuristic\_compute\_estimate(), mappel::ImageFormat2DBase::check\_image\_shape(), mappel::Gauss2DModel::←  
Stencil::compute\_derivatives(), mappel::Gauss2DsModel::Stencil::compute\_derivatives(), mappel::Gauss2DModel←  
::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::methods←  
::expected\_information(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsMAP::Gauss2DsMA←  
P(), mappel::Gauss2DsMLE::Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Image←  
Format2DBase::get\_num\_pixels(), mappel::ImageFormat2DBase::get\_size(), mappel::ImageFormat2DBase::get←  
\_stats(), mappel::methods::likelihood::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood←  
::debug::grad\_components(), mappel::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian←  
\_components(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(),  
mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh\_components(), mappel::Gauss2DModel←  
::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::←  
Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal←  
\_position(), mappel::ImageFormat2DBase::make\_image(), mappel::ImageFormat2DBase::make\_image\_stack(),  
mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DsModel::make\_internal\_1Dsum\_←  
estimator(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_←  
position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_prior\_normal←  
\_position(), mappel::methods::model\_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods←  
::likelihood::rllh(), mappel::methods::likelihood::debug::rllh\_components(), mappel::ImageFormat2DBase::set←  
size(), mappel::Gauss2DModel::set\_size(), mappel::Gauss2DsModel::set\_size(), mappel::methods::simulate\_image(),  
mappel::methods::simulate\_image\_from\_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2Ds←  
Model::Stencil::Stencil(), mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2Ds←  
Model::update\_internal\_1Dsum\_estimators().

#### 9.12.5.33 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.12.5.34 Gauss1DSumModelT mappel::Gauss2DModel::x\_model [protected],[inherited]

X-model fits 2D images X-axis (column sum)

Definition at line 105 of file Gauss2DModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_x(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::set\_psf\_sigma(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators().

#### 9.12.5.35 Gauss1DSumModelT mappel::Gauss2DModel::y\_model [protected],[inherited]

Y-model fits 2D images Y-axis (row sum)

Definition at line 106 of file Gauss2DModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::set\_psf\_sigma(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators().

The documentation for this class was generated from the following files:

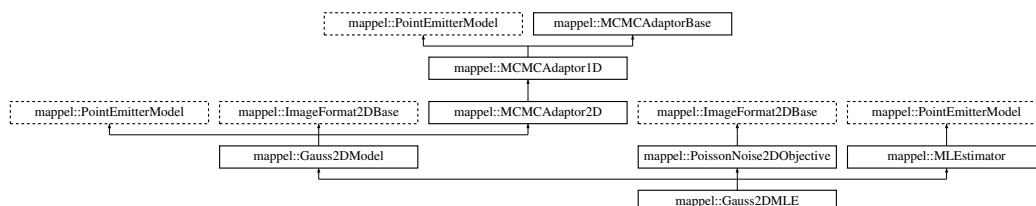
- [Gauss2DMP.h](#)
- [Gauss2DMP.cpp](#)

## 9.13 mappel::Gauss2DMLE Class Reference

A 2D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DMLE.h>
```

Inheritance diagram for mappel::Gauss2DMLE:



## Public Types

- using [Gauss1DSumModelT](#) = [Gauss1DMAP](#)
- using [StencilVecT](#) = `std::vector< Stencil >`
- using [ParamT](#) = `arma::vec`
- using [ParamVecT](#) = `arma::mat`
- using [ImageCoordT](#) = `uint32_t`
- using [ImagePixelT](#) = `double`
- template<class [CoordT](#) >  
using [ImageSizeShapeT](#) = `arma::Col< CoordT >`
- template<class [CoordT](#) >  
using [ImageSizeVecShapeT](#) = `arma::Mat< CoordT >`
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class [PixelT](#) >  
using [ImageShapeT](#) = `arma::Mat< PixelT >`
- template<class [PixelT](#) >  
using [ImageStackShapeT](#) = `arma::Cube< PixelT >`
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >
- using [ModelDataT](#) = [ImageT](#)
- using [ModelDataStackT](#) = [ImageStackT](#)

## Public Member Functions

- [Gauss2DMLE](#) ([ImageCoordT](#) [size](#), `double` [psf\\_sigma](#), `const std::string &prior_type=DefaultPriorType`)
- [Gauss2DMLE](#) (`const ImageSizeT &size`, `double` [psf\\_sigma](#), `const std::string &prior_type=DefaultPriorType`)
- [Gauss2DMLE](#) (`const ImageSizeT &size`, `const VecT &psf_sigma`, `const std::string &prior_type=DefaultPriorType`)
- [Gauss2DMLE](#) (`const ImageSizeT &size`, `const VecT &psf_sigma`, `CompositeDist &&prior`)
- [Gauss2DMLE](#) (`const ImageSizeT &size`, `const VecT &psf_sigma`, `const CompositeDist &prior`)
- [Gauss2DMLE](#) (`const Gauss2DMLE &o`)
- [Gauss2DMLE](#) & `operator=` (`const Gauss2DMLE &o`)
- [Gauss2DMLE](#) ([Gauss2DMLE](#) &&o)
- [Gauss2DMLE](#) & `operator=` ([Gauss2DMLE](#) &&o)
- `void` [set\\_hyperparams](#) (`const VecT &hyperparams`)
- `void` [set\\_prior](#) (`CompositeDist &&prior_`)
- `void` [set\\_prior](#) (`const CompositeDist &prior_`)
- `void` [set\\_size](#) (`const ImageSizeT &size_`)
- `const VecT &` [get\\_psf\\_sigma](#) (`()`) `const`
- `double` [get\\_psf\\_sigma](#) (`IdxT idx`) `const`
- `void` [set\\_psf\\_sigma](#) (`double new_psf_sigma`)
- `void` [set\\_psf\\_sigma](#) (`const VecT &new_psf_sigma`)
- [StatsT](#) [get\\_stats](#) (`()`) `const`
- [Stencil](#) [make\\_stencil](#) (`const ParamT &theta`, `bool compute_derivatives=true`) `const`  
*Make a new Model::Stencil object at theta.*
- `double` [pixel\\_model\\_value](#) (`int i`, `int j`, `const Stencil &s`) `const`
- `void` [pixel\\_grad](#) (`int i`, `int j`, `const Stencil &s`, [ParamT](#) &pgrad) `const`
- `void` [pixel\\_grad2](#) (`int i`, `int j`, `const Stencil &s`, [ParamT](#) &pgrad2) `const`
- `void` [pixel\\_hess](#) (`int i`, `int j`, `const Stencil &s`, [MatT](#) &hess) `const`



- void `pixel_hess_update` (int i, int j, const `Stencil` &s, double dm\_ratio\_m1, double dmm\_ratio, `ParamT` &grad, `MatT` &hess) const  
*pixel derivative inner loop calculations.*
- `Stencil initial_theta_estimate` (const `ImageT` &im) const  
*Fast, heuristic estimate of initial theta.*
- `Stencil initial_theta_estimate` (const `ImageT` &im, const `ParamT` &theta\_init) const
- `Stencil initial_theta_estimate` (const `ImageT` &im, const `ParamT` &theta\_init, const std::string &estimator) const
- `Gauss1DSumModelT debug_internal_sum_model_x` () const
- `Gauss1DSumModelT debug_internal_sum_model_y` () const
- `IdxT get_num_params` () const
- void `check_param_shape` (const `ParamT` &theta) const
- void `check_param_shape` (const `ParamVecT` &theta) const
- void `check_psf_sigma` (double `psf_sigma`) const
- void `check_psf_sigma` (const `VecT` &`psf_sigma`) const
- `ParamT make_param` () const
- template<class FillT >  
`ParamT make_param` (FillT fill) const
- `ParamVecT make_param_stack` (`IdxT` n) const
- template<class FillT >  
`ParamVecT make_param_stack` (`IdxT` n, FillT fill) const
- `MatT make_param_mat` () const
- template<class FillT >  
`MatT make_param_mat` (FillT fill) const
- `CubeT make_param_mat_stack` (`IdxT` n) const
- template<class FillT >  
`CubeT make_param_mat_stack` (`IdxT` n, FillT fill) const
- `CompositeDist & get_prior` ()
- const `CompositeDist & get_prior` () const
- `IdxT get_num_hyperparams` () const
- `VecT get_hyperparams` () const
- bool `has_hyperparam` (const std::string &name) const
- double `get_hyperparam_value` (const std::string &name) const
- int `get_hyperparam_index` (const std::string &name) const
- void `set_hyperparam_value` (const std::string &name, double value)
- void `rename_hyperparam` (const std::string &old\_name, const std::string &new\_name)
- `StringVecT get_param_names` () const
- void `set_param_names` (const `StringVecT` &desc)
- `StringVecT get_hyperparam_names` () const
- void `set_hyperparam_names` (const `StringVecT` &desc)
- template<class RngT >  
`ParamT sample_prior` (RngT &rng) const
- `ParamT sample_prior` () const
- void `set_bounds` (const `ParamT` &lbound, const `ParamT` &ubound)
- void `set_lbound` (const `ParamT` &lbound)
- void `set_ubound` (const `ParamT` &ubound)
- const `ParamT & get_lbound` () const
- const `ParamT & get_ubound` () const
- bool `theta_in_bounds` (const `ParamT` &theta) const
- void `bound_theta` (`ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT bounded_theta` (const `ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT reflected_theta` (const `ParamT` &theta) const

- [BoolVecT](#) [theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT](#) [bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT](#) [reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT](#) [make\\_image](#) () const
- [ImageStackT](#) [make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT](#) [get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT](#) [get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- template<class [ImT](#) >  
void [set\\_image\\_in\\_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const [ImT](#) &im) const
- const [ImageSizeT](#) & [get\\_size](#) () const
- [ImageCoordT](#) [get\\_size](#) ([IdxT](#) idx) const
- [ImageCoordT](#) [get\\_num\\_pixels](#) () const
- void [check\\_image\\_shape](#) (const [ImageT](#) &im) const  
*Check the shape of a single images is correct for model size.*
- void [check\\_image\\_shape](#) (const [ImageStackT](#) &ims) const  
*Check the shape of a stack of images is correct for model size.*
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, const [IdxVecT](#) &fixed\_parameters\_↵ mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)
- void [set\\_background\\_mcmc\\_sampling](#) (double eta\_bg=-1)
- void [set\\_mcmc\\_sigma\\_scale](#) (double scale)
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- [IdxT](#) [get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Member Functions

- static [CompositeDist](#) [make\\_default\\_prior](#) (const [ImageSizeT](#) &size, const std::string &prior\_type)
- static [CompositeDist](#) [make\\_default\\_prior\\_beta\\_position](#) (const [ImageSizeT](#) &size)
- static [CompositeDist](#) [make\\_default\\_prior\\_normal\\_position](#) (const [ImageSizeT](#) &size)
- static [CompositeDist](#) [make\\_prior\\_beta\\_position](#) (const [ImageSizeT](#) &size, double beta\_xpos, double beta\_ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static [CompositeDist](#) [make\\_prior\\_normal\\_position](#) (const [ImageSizeT](#) &size, double sigma\_xpos, double beta\_↵ ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static [prior\\_hessian::TruncatedNormalDist](#) [make\\_prior\\_component\\_position\\_normal](#) ([IdxT](#) size, double pos\_↵ sigma=[default\\_sigma\\_pos](#))
- static [prior\\_hessian::ScaledSymmetricBetaDist](#) [make\\_prior\\_component\\_position\\_beta](#) ([IdxT](#) size, double pos\_↵ beta=[default\\_beta\\_pos](#))
- static [prior\\_hessian::TruncatedGammaDist](#) [make\\_prior\\_component\\_intensity](#) (double mean=[default\\_mean\\_↵ l, double kappa=\[default\\\_intensity\\\_kappa\]\(#\)](#))
- static [prior\\_hessian::TruncatedParetoDist](#) [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵ sigma, double alpha=[default\\_alpha\\_sigma](#))
- static void [set\\_rng\\_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)

*Check the size argument for the model.*

## Static Public Attributes

- static const std::string `name`
- static const `StringVecT` `prior_types`
- static const std::string `DefaultPriorType` = "Normal"
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1
- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2
- static const `ImageCoordT` `num_dim` =2
- static const `ImageCoordT` `global_min_size` =3
- static const `ImageCoordT` `global_max_size` =512
- static const double `global_default_mcmc_sigma_scale` = 0.05
- static const double `global_max_mcmc_sigma_scale` = 0.5
- static const std::vector< std::string > `estimator_names`

## Protected Member Functions

- void `update_internal_1Dsum_estimators` ()
- void `set_mcmc_num_phases` (`IdxT` num\_phases)

## Static Protected Member Functions

- static `Gauss1DSumModelT` `make_internal_1Dsum_estimator` (`IdxT` dim, const `ImageSizeT` &size, const `VecT` &psf\_sigma, const `CompositeDist` &prior)

## Protected Attributes

- `VecT` `psf_sigma`
- `Gauss1DSumModelT` `x_model`
- `Gauss1DSumModelT` `y_model`
- `CompositeDist` `prior`
- `IdxT` `num_params`
- `IdxT` `num_hyperparams`
- `ParamT` `lbound`
- `ParamT` `ubound`
- `ImageSizeT` `size`
- double `eta_y` =0
- double `eta_x` =0
- double `eta_l` =0
- double `eta_bg` =0
- `IdxT` `num_phases`
- double `sigma_scale`

### 9.13.1 Detailed Description

A 2D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective.

Model: [Gauss2DModel](#) - 2D Gaussian PSF with fixed PSF sigma Objective: [PoissonNoise2DObjective](#) - Poisson noise model for 2D Estimator: [MLEstimator](#) - Pure-likelihood estimator

Definition at line 24 of file Gauss2DMLE.h.

### 9.13.2 Member Typedef Documentation

**9.13.2.1** `using mappel::Gauss2DModel::Gauss1DSumModelT = Gauss1DMap` [inherited]

Definition at line 23 of file Gauss2DModel.h.

**9.13.2.2** `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

**9.13.2.3** `using mappel::ImageFormat2DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

**9.13.2.4** `template<class PixelT > using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>`  
[inherited]

Shape of the data type for a single image

Definition at line 32 of file ImageFormat2DBase.h.

**9.13.2.5** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>`  
[inherited]

Shape of the data type to store a single image's coordinates

Definition at line 27 of file ImageFormat2DBase.h.

**9.13.2.6** `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`  
[inherited]

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

9.13.2.7 `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>`  
`[inherited]`

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

9.13.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

9.13.2.9 `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

9.13.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

9.13.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

9.13.2.12 `using mappel::PoissonNoise2DObjective::ModelDataStackT = ImageStackT` `[inherited]`

Objective function data stack type: 2D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise2DObjective.h.

9.13.2.13 `using mappel::PoissonNoise2DObjective::ModelDataT = ImageT` `[inherited]`

Objective function data type: 2D double precision image, gain-corrected to approximate photons counts

Definition at line 24 of file PoissonNoise2DObjective.h.

9.13.2.14 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.13.2.15 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

9.13.2.16 `using mappel::Gauss2DModel::StencilVecT = std::vector<Stencil>` [inherited]

Definition at line 47 of file Gauss2DModel.h.

### 9.13.3 Constructor & Destructor Documentation

9.13.3.1 `mappel::Gauss2DMLE::Gauss2DMLE ( ImageCoordT size, double psf_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 11 of file Gauss2DMLE.cpp.

9.13.3.2 `mappel::Gauss2DMLE::Gauss2DMLE ( const ImageSizeT & size, double psf_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 15 of file Gauss2DMLE.cpp.

9.13.3.3 `mappel::Gauss2DMLE::Gauss2DMLE ( const ImageSizeT & size, const VecT & psf_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 19 of file Gauss2DMLE.cpp.

9.13.3.4 `mappel::Gauss2DMLE::Gauss2DMLE ( const ImageSizeT & size, const VecT & psf_sigma, CompositeDist && prior )`

Definition at line 23 of file Gauss2DMLE.cpp.

9.13.3.5 `mappel::Gauss2DMLE::Gauss2DMLE ( const ImageSizeT & size, const VecT & psf_sigma, const CompositeDist & prior )`

Definition at line 31 of file Gauss2DMLE.cpp.

9.13.3.6 `mappel::Gauss2DMLE::Gauss2DMLE ( const Gauss2DMLE & o )`

Definition at line 39 of file Gauss2DMLE.cpp.

## 9.13.3.7 mappel::Gauss2DMLE::Gauss2DMLE ( Gauss2DMLE &amp;&amp; o )

Definition at line 47 of file Gauss2DMLE.cpp.

## 9.13.4 Member Function Documentation

## 9.13.4.1 void mappel::PointEmitterModel::bound\_theta ( ParamT &amp; theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

## 9.13.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT &amp; theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

## 9.13.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT &amp; theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

## 9.13.4.4 void mappel::ImageFormat2DBase::check\_image\_shape ( const ImageT &amp; im ) const [inherited]

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

## 9.13.4.5 void mappel::ImageFormat2DBase::check\_image\_shape ( const ImageStackT &amp; ims ) const [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.13.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` `[inherited]`

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

**9.13.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

**9.13.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` `[inherited]`

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

**9.13.4.9** `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

**9.13.4.10** `void mappel::ImageFormat2DBase::check_size ( const ImageSizeT & size )` `[static],[inherited]`

Check the size argument for the model.

Definition at line 60 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::global\_max\_size, and mappel::ImageFormat2DBase::global\_min\_size.

Referenced by mappel::ImageFormat2DBase::ImageFormat2DBase(), and mappel::ImageFormat2DBase::set\_size().

**9.13.4.11** `Gauss1DSumModelT mappel::Gauss2DModel::debug_internal_sum_model_x ( ) const` `[inline],[inherited]`

Definition at line 89 of file Gauss2DModel.h.

References mappel::Gauss2DModel::x\_model.



**9.13.4.12** `Gauss1DSumModelT` `mappel::Gauss2DModel::debug_internal_sum_model_y ( ) const` `[inline]`,  
`[inherited]`

Definition at line 90 of file `Gauss2DModel.h`.

References `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DModel::operator=()`, `mappel::PointEmitterModel::prior`, `mappel::Gauss2DModel::psf_sigma`, `mappel::ImageFormat2DBase::size`, `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DModel::y_model`.

**9.13.4.13** `int` `mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 236 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.13.4.14** `StringVecT` `mappel::PointEmitterModel::get_hyperparam_names ( ) const` `[inline]`, `[inherited]`

Definition at line 256 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.13.4.15** `double` `mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 232 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.13.4.16** `PointEmitterModel::ParamT` `mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.13.4.17** `ImageFormat2DBase::ImageT` `mappel::ImageFormat2DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` `[inline]`, `[inherited]`

Definition at line 106 of file `ImageFormat2DBase.h`.

**9.13.4.18** `const PointEmitterModel::ParamT &` `mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.13.4.19** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

**9.13.4.20** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

**9.13.4.21** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

**9.13.4.22** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.13.4.23** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_num_pixels ( ) const` `[inline],[inherited]`

Definition at line 79 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

**9.13.4.24** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.13.4.25** `CompositeDist & mappel::PointEmitterModel::get_prior ( )` `[inline],[inherited]`

Definition at line 200 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

9.13.4.26 `const CompositeDist & mappel::PointEmitterModel::get_prior ( ) const` `[inline],[inherited]`

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.13.4.27 `const VecT & mappel::Gauss2DModel::get_psf_sigma ( ) const` `[inline],[inherited]`

Definition at line 142 of file `Gauss2DModel.h`.

References `mappel::Gauss2DModel::psf_sigma`.

Referenced by `mappel::Gauss2DModel::get_stats()`.

9.13.4.28 `double mappel::Gauss2DModel::get_psf_sigma ( IdxT idx ) const` `[inherited]`

Definition at line 132 of file `Gauss2DModel.cpp`.

References `mappel::Gauss2DModel::DefaultPriorType`, `mappel::Gauss2DModel::prior_types`, and `mappel::Gauss2DModel::psf_sigma`.

9.13.4.29 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ( )` `[static],[inherited]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

9.13.4.30 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ( )` `[static],[inherited]`

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

9.13.4.31 `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size ( ) const` `[inline],[inherited]`

Definition at line 74 of file `ImageFormat2DBase.h`.

References `mappel::ImageFormat2DBase::size`.

9.13.4.32 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size ( IdxT idx ) const` `[inherited]`

Definition at line 41 of file `ImageFormat2DBase.cpp`.

References `mappel::ImageFormat2DBase::size`.

**9.13.4.33 ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::get\_size\_image\_stack ( const ImageStackT & stack ) const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

**9.13.4.34 StatsT** mappel::Gauss2DModel::get\_stats ( ) const [inherited]

Definition at line 268 of file Gauss2DModel.cpp.

References mappel::Gauss2DModel::get\_psf\_sigma(), mappel::MCMCAdaptor2D::get\_stats(), mappel::ImageFormat2DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

**9.13.4.35 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const** [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

**9.13.4.36 bool** mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.13.4.37 Gauss2DModel::Stencil** mappel::Gauss2DModel::initial\_theta\_estimate ( const ImageT & im ) const [inline],[inherited]

Fast, heuristic estimate of initial theta.

Definition at line 194 of file Gauss2DModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::PointEmitterModel::make\_param().

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate().

**9.13.4.38 Gauss2DModel::Stencil** mappel::Gauss2DModel::initial\_theta\_estimate ( const ImageT & im, const ParamT & theta\_init ) const [inline],[inherited]

Definition at line 201 of file Gauss2DModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::Gauss2DModel::initial\_theta\_estimate().

**9.13.4.39** `Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init, const std::string & estimator ) const` `[inherited]`

Definition at line 303 of file Gauss2DModel.cpp.

References `mappel::Gauss2DModel::Stencil::bg()`, `mappel::methods::estimate_max()`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::PointEmitterModel::lbound`, `mappel::Gauss2DModel::make_stencil()`, `mappel::PointEmitterModel::num_params`, `mappel::ImageFormat2DBase::size`, `mappel::estimator::MLEData::theta`, `mappel::PointEmitterModel::theta_in_bounds()`, `mappel::PointEmitterModel::ubound`, `mappel::Gauss2DModel::x_model`, and `mappel::Gauss2DModel::y_model`.

**9.13.4.40** `CompositeDist mappel::Gauss2DModel::make_default_prior ( const ImageSizeT & size, const std::string & prior_type )` `[static]`, `[inherited]`

Definition at line 150 of file Gauss2DModel.cpp.

References `mappel::istarts_with()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, and `mappel::Gauss2DModel::make_default_prior_normal_position()`.

**9.13.4.41** `CompositeDist mappel::Gauss2DModel::make_default_prior_beta_position ( const ImageSizeT & size )` `[static]`, `[inherited]`

Definition at line 171 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::Gauss2DModel::make_default_prior()`.

**9.13.4.42** `CompositeDist mappel::Gauss2DModel::make_default_prior_normal_position ( const ImageSizeT & size )` `[static]`, `[inherited]`

Definition at line 182 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::Gauss2DModel::make_default_prior()`.

**9.13.4.43** `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image ( ) const` `[inline]`, `[inherited]`

Definition at line 85 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

**9.13.4.44 ImageFormat2DBase::ImageStackT** mappel::ImageFormat2DBase::make\_image\_stack ( ImageCoordT *n* ) const  
[inline],[inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.13.4.45 Gauss2DModel::Gauss1DSumModelT** mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator ( IdxT *dim*,  
const ImageSizeT & *size*, const VecT & *psf\_sigma*, const CompositeDist & *prior* ) [static],[protected],  
[inherited]

Definition at line 62 of file Gauss2DModel.cpp.

References mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_normal\_↵  
position(), mappel::Gauss2DModel::psf\_sigma, and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(),  
and mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators().

**9.13.4.46 PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline],[inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(),  
mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::↵  
Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel↵  
::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_↵  
update().

**9.13.4.47 template<class FillT > PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( FillT *fill* ) const  
[inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.13.4.48 MatT** mappel::PointEmitterModel::make\_param\_mat ( ) const [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.13.4.49 template<class FillT > MatT** mappel::PointEmitterModel::make\_param\_mat ( FillT *fill* ) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.13.4.50 **CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( *IdxT n* ) const [inline], [inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.13.4.51 **template<class FillT > CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( *IdxT n*, *FillT fill* ) const [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.13.4.52 **PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make\_param\_stack ( *IdxT n* ) const [inline], [inherited]

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.13.4.53 **template<class FillT > PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make\_param\_stack ( *IdxT n*, *FillT fill* ) const [inherited]

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.13.4.54 **CompositeDist** mappel::Gauss2DModel::make\_prior\_beta\_position ( const *ImageSizeT & size*, double *beta\_xpos*, double *beta\_ypos*, double *mean\_l*, double *kappa\_l*, double *mean\_bg*, double *kappa\_bg* ) [static], [inherited]

Definition at line 193 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), and mappel::ImageFormat2DBase::size.

9.13.4.55 **prior\_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make\_prior\_component\_intensity ( double *mean = default\_mean\_l*, double *kappa = default\_intensity\_kappa* ) [static], [inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_l.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.13.4.56 **prior\_hessian::ScaledSymmetricBetaDist** **mappel::PointEmitterModel::make\_prior\_component\_position\_beta** ( *IdxT size*, *double pos\_beta = default\_beta\_pos* ) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

9.13.4.57 **prior\_hessian::TruncatedNormalDist** **mappel::PointEmitterModel::make\_prior\_component\_position\_normal** ( *IdxT size*, *double pos\_sigma = default\_sigma\_pos* ) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.13.4.58 **prior\_hessian::TruncatedParetoDist** **mappel::PointEmitterModel::make\_prior\_component\_sigma** ( *double min\_sigma*, *double max\_sigma*, *double alpha = default\_alpha\_sigma* ) [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.13.4.59 **CompositeDist** **mappel::Gauss2DModel::make\_prior\_normal\_position** ( *const ImageSizeT & size*, *double sigma\_xpos*, *double beta\_ypos*, *double mean\_l*, *double kappa\_l*, *double mean\_bg*, *double kappa\_bg* ) [static],[inherited]

Definition at line 206 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::ImageFormat2DBase::size`.

9.13.4.60 **Gauss2DModel::Stencil** **mappel::Gauss2DModel::make\_stencil** ( *const ParamT & theta*, *bool compute\_derivatives = true* ) **const** [inline],[inherited]

Make a new `Model::Stencil` object at `theta`.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular `theta` (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the `mappel::methods` namespace accept a `const Stencil` reference in place of the model parameter.

Throws `mappel::ModelBoundsError` if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`



## Parameters

<i>theta</i>	Prameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

## Returns

A new [Stencil](#) object ready to compute with

Definition at line 131 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`.

#### 9.13.4.61 Gauss2DMLE & mappel::Gauss2DMLE::operator= ( const Gauss2DMLE & o )

Definition at line 55 of file Gauss2DMLE.cpp.

References `mappel::MLEstimator::operator=()`, `mappel::PoissonNoise2DObjective::operator=()`, `mappel::Image↔Format2DBase::operator=()`, `mappel::Gauss2DModel::operator=()`, and `mappel::PointEmitterModel::operator=()`.

#### 9.13.4.62 Gauss2DMLE & mappel::Gauss2DMLE::operator= ( Gauss2DMLE && o )

Definition at line 66 of file Gauss2DMLE.cpp.

References `mappel::MLEstimator::operator=()`, `mappel::PoissonNoise2DObjective::operator=()`, `mappel::Image↔Format2DBase::operator=()`, `mappel::Gauss2DModel::operator=()`, and `mappel::PointEmitterModel::operator=()`.

#### 9.13.4.63 void mappel::Gauss2DModel::pixel\_grad ( int i, int j, const Stencil & s, ParamT & pgrad ) const [inline], [inherited]

Definition at line 159 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::↔Stencil::I()`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

Referenced by `mappel::Gauss2DModel::pixel_hess_update()`.

#### 9.13.4.64 void mappel::Gauss2DModel::pixel\_grad2 ( int i, int j, const Stencil & s, ParamT & pgrad2 ) const [inline], [inherited]

Definition at line 170 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::↔Stencil::I()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::↔Stencil::Y`.

**9.13.4.65** `void mappel::Gauss2DModel::pixel_hess ( int i, int j, const Stencil & s, MatT & hess ) const` `[inline]`,  
`[inherited]`

Definition at line 181 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

**9.13.4.66** `void mappel::Gauss2DModel::pixel_hess_update ( int i, int j, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess ) const` `[inherited]`

pixel derivative inner loop calculations.

Definition at line 282 of file Gauss2DModel.cpp.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::PointEmitterModel::make_param()`, `mappel::Gauss2DModel::pixel_grad()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

**9.13.4.67** `double mappel::Gauss2DModel::pixel_model_value ( int i, int j, const Stencil & s ) const` `[inline]`,  
`[inherited]`

Definition at line 152 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::bg()`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

**9.13.4.68** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const` `[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.13.4.69** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.13.4.70** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )` `[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.13.4.71 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const` [inherited]

Definition at line 59 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_`  
`x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.13.4.72 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const` [inherited]

Definition at line 74 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_`  
`x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.13.4.73 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const` [inherited]

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.13.4.74 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` [inline],[inherited]

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.13.4.75 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::Point`  
`EmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_`  
`ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.13.4.76 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitter`  
`Model::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.13.4.77 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc ) [inline], [inherited]`

Definition at line 260 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.13.4.78 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.13.4.79 `void mappel::Gauss2DModel::set_hyperparams ( const VecT & hyperparams ) [inherited]`

Definition at line 109 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::set\_hyperparams(), and mappel::Gauss2DModel::update\_internal\_1Dsum\_↵ estimators().

9.13.4.80 `template<class ImT > void mappel::ImageFormat2DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]`

Definition at line 113 of file ImageFormat2DBase.h.

9.13.4.81 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 ) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_mean\_I, mappel::MCMCAdaptor1D::eta\_I, mappel::PointEmitter↵ Model::get\_hyperparam\_value(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

9.13.4.82 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitter↵ Model::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.13.4.83 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases ) [protected], [inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

Referenced by mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), and mappel::MCMCAdaptor2Ds::MCMCAdaptor2↵ Ds().

9.13.4.84 void mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale ( double *scale* ) [inherited]

Definition at line 39 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::global\_max\_mcmc\_sigma\_scale, and mappel::MCMCAdaptorBase::sigma\_↵\_scale.

9.13.4.85 void mappel::PointEmitterModel::set\_param\_names ( const StringVecT & *desc* ) [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.13.4.86 void mappel::Gauss2DModel::set\_prior ( CompositeDist && *prior\_* ) [inherited]

Definition at line 97 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::set\_prior(), and mappel::Gauss2DModel::update\_internal\_1Dsum\_↵estimators().

9.13.4.87 void mappel::Gauss2DModel::set\_prior ( const CompositeDist & *prior\_* ) [inherited]

Definition at line 103 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::set\_prior(), and mappel::Gauss2DModel::update\_internal\_1Dsum\_↵estimators().

9.13.4.88 void mappel::Gauss2DModel::set\_psf\_sigma ( double *new\_psf\_sigma* ) [inline],[inherited]

Definition at line 146 of file Gauss2DModel.h.

9.13.4.89 void mappel::Gauss2DModel::set\_psf\_sigma ( const VecT & *new\_psf\_sigma* ) [inherited]

Definition at line 123 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::check\_psf\_sigma(), mappel::Gauss2DModel::psf\_sigma, mappel::Gauss1D\_↵Model::set\_psf\_sigma(), mappel::Gauss2DModel::x\_model, and mappel::Gauss2DModel::y\_model.

9.13.4.90 void mappel::PointEmitterModel::set\_rng\_seed ( RngSeedT *seed* ) [static],[inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.13.4.91 void mappel::Gauss2DModel::set\_size ( const ImageSizeT & *size\_* ) [inherited]

Definition at line 115 of file Gauss2DModel.cpp.

References mappel::ImageFormat2DBase::set\_size(), mappel::ImageFormat1DBase::set\_size(), mappel::Image\_↵Format2DBase::size, mappel::Gauss2DModel::x\_model, and mappel::Gauss2DModel::y\_model.

**9.13.4.92** `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

**9.13.4.93** `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const` [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.13.4.94** `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const` [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.13.4.95** `void mappel::Gauss2DModel::update_internal_1Dsum_estimators ( )` [protected],[inherited]

Definition at line 91 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::get_prior()`, `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::ImageFormat2DBase::size`, `mappel::Gauss2DModel::x_model`, and `mappel::Gauss2DModel::y_model`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DModel::set_prior()`.

## 9.13.5 Member Data Documentation

**9.13.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static],[inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

9.13.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.13.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.13.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

9.13.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.13.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` [static],[inherited]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.13.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` [static],[inherited]

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.13.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.13.5.9** `const std::string mappel::Gauss2DModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 51 of file Gauss2DModel.h.

Referenced by `mappel::Gauss2DModel::get_psf_sigma()`.

**9.13.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.13.5.11** `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` `[static], [inherited]`

Definition at line 23 of file PoissonNoise2DObjective.h.

**9.13.5.12** `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.13.5.13** `double mappel::MCMCAdaptor1D::eta_I = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_I` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.



**9.13.5.14** `double mappel::MCMCAdaptor1D::eta_x=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to theta\_x in the random walk MCMC sampling

Definition at line 31 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

**9.13.5.15** `double mappel::MCMCAdaptor2D::eta_y=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to theta\_y in the random walk MCMC sampling

Definition at line 28 of file MCMCAdaptor2D.h.

Referenced by `mappel::MCMCAdaptor2D::get_stats()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2D::operator=()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`.

**9.13.5.16** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static],[inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.13.5.17** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static],[inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.13.5.18** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static],[inherited]`

Global maximum for any psf\_sigma. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.13.5.19** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size=512` `[static],[inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.13.5.20** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.13.5.21** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file `ImageFormat2DBase.h`.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.13.5.22** `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.13.5.23** `const std::string mappel::Gauss2DMLE::name` `[static]`

Definition at line 37 of file `Gauss2DMLE.h`.

**9.13.5.24** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim = 2` `[static], [inherited]`

Number of image dimensions.

Definition at line 37 of file `ImageFormat2DBase.h`.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

**9.13.5.25** `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

#### 9.13.5.26 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.13.5.27 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

#### 9.13.5.28 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.13.5.29 `const StringVecT mappel::Gauss2DModel::prior_types` `[static]`, `[inherited]`

**Initial value:**

```
= { "Beta",
    "Normal"
}
```

Definition at line 50 of file `Gauss2DModel.h`.

Referenced by `mappel::Gauss2DModel::get_psf_sigma()`.

### 9.13.5.30 VecT mappel::Gauss2DModel::psf\_sigma [protected],[inherited]

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 104 of file Gauss2DModel.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_←  
\_heuristic\_compute\_estimate(), mappel::Gauss2DModel::Stencil::compute\_derivatives(), mappel::Gauss2DModel←  
::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DModel::get\_psf\_←  
sigma(), mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DModel::operator=(), mappel←  
::Gauss2DModel::pixel\_grad2(), mappel::Gauss2DModel::pixel\_hess(), mappel::Gauss2DModel::pixel\_hess\_update(),  
mappel::Gauss2DModel::set\_psf\_sigma(), mappel::Gauss2DModel::Stencil::Stencil(), and mappel::Gauss2DModel←  
::update\_internal\_1Dsum\_estimators().

### 9.13.5.31 double mappel::MCMCAdaptorBase::sigma\_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(),  
mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMC←  
Adaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set←  
\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMC←  
AdaptorBase::set\_mcmc\_sigma\_scale().

### 9.13.5.32 ImageSizeT mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_←  
\_heuristic\_compute\_estimate(), mappel::ImageFormat2DBase::check\_image\_shape(), mappel::Gauss2DModel::←  
Stencil::compute\_derivatives(), mappel::Gauss2DsModel::Stencil::compute\_derivatives(), mappel::Gauss2DModel←  
::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::methods←  
::expected\_information(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsMAP::Gauss2DsMA←  
P(), mappel::Gauss2DsMLE::Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Image←  
Format2DBase::get\_num\_pixels(), mappel::ImageFormat2DBase::get\_size(), mappel::ImageFormat2DBase::get←  
\_stats(), mappel::methods::likelihood::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood←  
::debug::grad\_components(), mappel::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian←  
\_components(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(),  
mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh\_components(), mappel::Gauss2DModel←  
::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::←  
Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal←  
\_position(), mappel::ImageFormat2DBase::make\_image(), mappel::ImageFormat2DBase::make\_image\_stack(),  
mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DsModel::make\_internal\_1Dsum\_←  
estimator(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_←  
position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_prior\_normal←  
\_position(), mappel::methods::model\_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods←  
::likelihood::rllh(), mappel::methods::likelihood::debug::rllh\_components(), mappel::ImageFormat2DBase::set←  
size(), mappel::Gauss2DModel::set\_size(), mappel::Gauss2DsModel::set\_size(), mappel::methods::simulate\_image(),  
mappel::methods::simulate\_image\_from\_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2Ds←  
Model::Stencil::Stencil(), mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2Ds←  
Model::update\_internal\_1Dsum\_estimators().

#### 9.13.5.33 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.13.5.34 Gauss1DSumModelT mappel::Gauss2DModel::x\_model [protected],[inherited]

X-model fits 2D images X-axis (column sum)

Definition at line 105 of file Gauss2DModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_x(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::set\_psf\_sigma(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators().

#### 9.13.5.35 Gauss1DSumModelT mappel::Gauss2DModel::y\_model [protected],[inherited]

Y-model fits 2D images Y-axis (row sum)

Definition at line 106 of file Gauss2DModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::set\_psf\_sigma(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators().

The documentation for this class was generated from the following files:

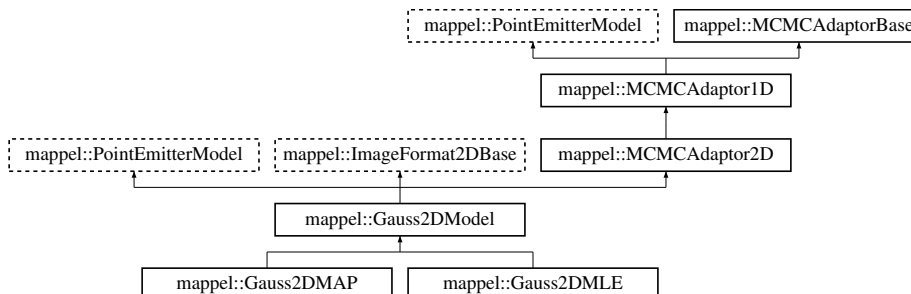
- [Gauss2DMLE.h](#)
- [Gauss2DMLE.cpp](#)

## 9.14 mappel::Gauss2DModel Class Reference

A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DModel.h>
```

Inheritance diagram for mappel::Gauss2DModel:



## Classes

- class [Stencil](#)  
*Stencil for 2D fixed-sigma models.*

## Public Types

- using [Gauss1DSumModelT](#) = [Gauss1DMP](#)
- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >

## Public Member Functions

- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- const [VecT](#) & [get\\_psf\\_sigma](#) () const
- double [get\\_psf\\_sigma](#) (IdxT idx) const
- void [set\\_psf\\_sigma](#) (double new\_psf\_sigma)
- void [set\\_psf\\_sigma](#) (const [VecT](#) &new\_psf\_sigma)
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &theta, bool compute\_derivatives=true) const  
*Make a new Model::Stencil object at theta.*
- double [pixel\\_model\\_value](#) (int i, int j, const [Stencil](#) &s) const
- void [pixel\\_grad](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel\\_grad2](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel\\_hess](#) (int i, int j, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel\\_hess\\_update](#) (int i, int j, const [Stencil](#) &s, double dm\_ratio\_m1, double dmm\_ratio, [ParamT](#) &grad, [MatT](#) &hess) const  
*pixel derivative inner loop calculations.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im) const  
*Fast, heuristic estimate of initial theta.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init) const

- [Stencil initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init, const std::string &estimator) const
- [Gauss1DSumModelT debug\\_internal\\_sum\\_model\\_x](#) () const
- [Gauss1DSumModelT debug\\_internal\\_sum\\_model\\_y](#) () const
- [IdxT get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double [psf\\_sigma](#)) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &[psf\\_sigma](#)) const
- [ParamT make\\_param](#) () const
- template<class FillT >  
  [ParamT make\\_param](#) (FillT fill) const
- [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT make\\_param\\_mat](#) () const
- template<class FillT >  
  [MatT make\\_param\\_mat](#) (FillT fill) const
- [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- [CompositeDist & get\\_prior](#) ()
- const [CompositeDist & get\\_prior](#) () const
- [IdxT get\\_num\\_hyperparams](#) () const
- [VecT get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT sample\\_prior](#) (RngT &rng) const
- [ParamT sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT & get\\_lbound](#) () const
- const [ParamT & get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT make\\_image](#) () const
- [ImageStackT make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const

- `ImageT get_image_from_stack` (const `ImageStackT` &stack, `ImageCoordT` n) const
- `template<class ImT >`  
void `set_image_in_stack` (`ImageStackT` &stack, `ImageCoordT` n, const `ImT` &im) const
- const `ImageSizeT` & `get_size` () const
- `ImageCoordT` `get_size` (`IdxT` idx) const
- `ImageCoordT` `get_num_pixels` () const
- void `check_image_shape` (const `ImageT` &im) const  
*Check the shape of a single images is correct for model size.*
- void `check_image_shape` (const `ImageStackT` &ims) const  
*Check the shape of a stack of images is correct for model size.*
- void `sample_mcmc_candidate` (`IdxT` sample\_index, `ParamT` &candidate, double step\_scale=1.0) const
- void `sample_mcmc_candidate` (`IdxT` sample\_index, `ParamT` &candidate, const `IdxVecT` &fixed\_parameters\_↵  
mask, double step\_scale=1.0) const
- void `set_intensity_mcmc_sampling` (double eta\_l=-1)
- void `set_background_mcmc_sampling` (double eta\_bg=-1)
- void `set_mcmc_sigma_scale` (double scale)
- double `get_mcmc_sigma_scale` () const
- `IdxT` `get_mcmc_num_phases` () const

#### Static Public Member Functions

- static CompositeDist `make_default_prior` (const `ImageSizeT` &size, const std::string &prior\_type)
  - static CompositeDist `make_default_prior_beta_position` (const `ImageSizeT` &size)
  - static CompositeDist `make_default_prior_normal_position` (const `ImageSizeT` &size)
  - static CompositeDist `make_prior_beta_position` (const `ImageSizeT` &size, double beta\_xpos, double beta\_ypos,  
double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
  - static CompositeDist `make_prior_normal_position` (const `ImageSizeT` &size, double sigma\_xpos, double beta\_↵  
ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
  - static prior\_hessian::TruncatedNormalDist `make_prior_component_position_normal` (`IdxT` size, double pos\_↵  
sigma=default\_sigma\_pos)
  - static prior\_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta` (`IdxT` size, double pos\_↵  
beta=default\_beta\_pos)
  - static prior\_hessian::TruncatedGammaDist `make_prior_component_intensity` (double mean=default\_mean\_↵  
l, double kappa=default\_intensity\_kappa)
  - static prior\_hessian::TruncatedParetoDist `make_prior_component_sigma` (double min\_sigma, double max\_↵  
sigma, double alpha=default\_alpha\_sigma)
  - static void `set_rng_seed` (`RngSeedT` seed)
  - static `ParallelRngManagerT` & `get_rng_manager` ()
  - static `ParallelRngGeneratorT` & `get_rng_generator` ()
  - static void `check_size` (const `ImageSizeT` &size\_)
- Check the size argument for the model.*

#### Static Public Attributes

- static const `StringVecT` `prior_types`
- static const std::string `DefaultPriorType` = "Normal"
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1



- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) =2
- static const [ImageCoordT](#) [global\\_min\\_size](#) =3
- static const [ImageCoordT](#) [global\\_max\\_size](#) =512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5

#### Protected Member Functions

- [Gauss2DModel](#) (const [ImageSizeT](#) &size, const [VecT](#) &psf\_sigma)
- [Gauss2DModel](#) (const [Gauss2DModel](#) &o)
- [Gauss2DModel](#) ([Gauss2DModel](#) &&o)
- [Gauss2DModel](#) & operator= (const [Gauss2DModel](#) &o)
- [Gauss2DModel](#) & operator= ([Gauss2DModel](#) &&o)
- void [update\\_internal\\_1Dsum\\_estimators](#) ()
- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) num\_phases)

#### Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make\\_internal\\_1Dsum\\_estimator](#) ([IdxT](#) dim, const [ImageSizeT](#) &size, const [VecT](#) &psf\_sigma, const [CompositeDist](#) &prior)

#### Protected Attributes

- [VecT](#) [psf\\_sigma](#)
- [Gauss1DSumModelT](#) [x\\_model](#)
- [Gauss1DSumModelT](#) [y\\_model](#)
- [CompositeDist](#) [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta\\_y](#) =0
- double [eta\\_x](#) =0
- double [eta\\_l](#) =0
- double [eta\\_bg](#) =0
- [IdxT](#) [num\\_phases](#)
- double [sigma\\_scale](#)

### 9.14.1 Detailed Description

A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma.

Definition at line 20 of file Gauss2DModel.h.

### 9.14.2 Member Typedef Documentation

#### 9.14.2.1 `using mappel::Gauss2DModel::Gauss1DSumModelT = Gauss1DMap`

Definition at line 23 of file Gauss2DModel.h.

#### 9.14.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

#### 9.14.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

#### 9.14.2.4 `template<class PixelT > using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>` [inherited]

Shape of the data type for a single image

Definition at line 32 of file ImageFormat2DBase.h.

#### 9.14.2.5 `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>` [inherited]

Shape of the data type to store a single image's coordinates

Definition at line 27 of file ImageFormat2DBase.h.

#### 9.14.2.6 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

9.14.2.7 `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>`  
[inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

9.14.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
[inherited]

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

9.14.2.9 `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`  
[inherited]

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

9.14.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
[inherited]

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

9.14.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

9.14.2.12 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.14.2.13 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

9.14.2.14 `using mappel::Gauss2DModel::StencilVecT = std::vector<Stencil>`

Definition at line 47 of file Gauss2DModel.h.

### 9.14.3 Constructor & Destructor Documentation

#### 9.14.3.1 `mappel::Gauss2DModel::Gauss2DModel ( const ImageSizeT & size, const VecT & psf_sigma )` [protected]

Definition at line 12 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `make_internal_1Dsum_estimator()`, `mappel::PointEmitterModel::prior`, `psf_sigma`, `mappel::ImageFormat2DBase::size`, and `y_model`.

Referenced by `debug_internal_sum_model_y()`.

#### 9.14.3.2 `mappel::Gauss2DModel::Gauss2DModel ( const Gauss2DModel & o )` [protected]

Definition at line 22 of file Gauss2DModel.cpp.

References `make_internal_1Dsum_estimator()`, `mappel::PointEmitterModel::prior`, `psf_sigma`, `mappel::ImageFormat2DBase::size`, and `y_model`.

#### 9.14.3.3 `mappel::Gauss2DModel::Gauss2DModel ( Gauss2DModel && o )` [protected]

Definition at line 30 of file Gauss2DModel.cpp.

References `make_internal_1Dsum_estimator()`, `mappel::PointEmitterModel::prior`, `psf_sigma`, `mappel::ImageFormat2DBase::size`, and `y_model`.

### 9.14.4 Member Function Documentation

#### 9.14.4.1 `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

#### 9.14.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

9.14.4.3 **PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & *theta*, double *epsilon* = bounds\_epsilon ) const** [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

9.14.4.4 **void mappel::ImageFormat2DBase::check\_image\_shape ( const ImageT & *im* ) const** [inherited]

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

9.14.4.5 **void mappel::ImageFormat2DBase::check\_image\_shape ( const ImageStackT & *ims* ) const** [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

9.14.4.6 **void mappel::PointEmitterModel::check\_param\_shape ( const ParamT & *theta* ) const** [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.14.4.7 **void mappel::PointEmitterModel::check\_param\_shape ( const ParamVecT & *theta* ) const** [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

9.14.4.8 **void mappel::PointEmitterModel::check\_psf\_sigma ( double *psf\_sigma* ) const** [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and set\_psf\_sigma().

9.14.4.9 **void mappel::PointEmitterModel::check\_psf\_sigma ( const VecT & psf\_sigma ) const** [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

9.14.4.10 **void mappel::ImageFormat2DBase::check\_size ( const ImageSizeT & size\_ )** [static],[inherited]

Check the size argument for the model.

Definition at line 60 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::global\_max\_size, and mappel::ImageFormat2DBase::global\_min\_size.

Referenced by mappel::ImageFormat2DBase::ImageFormat2DBase(), and mappel::ImageFormat2DBase::set\_size().

9.14.4.11 **Gauss1DSumModelT mappel::Gauss2DModel::debug\_internal\_sum\_model\_x ( ) const** [inline]

Definition at line 89 of file Gauss2DModel.h.

References x\_model.

9.14.4.12 **Gauss1DSumModelT mappel::Gauss2DModel::debug\_internal\_sum\_model\_y ( ) const** [inline]

Definition at line 90 of file Gauss2DModel.h.

References Gauss2DModel(), make\_internal\_1Dsum\_estimator(), operator=(), mappel::PointEmitterModel::prior, psf\_sigma, mappel::ImageFormat2DBase::size, update\_internal\_1Dsum\_estimators(), and y\_model.

9.14.4.13 **int mappel::PointEmitterModel::get\_hyperparam\_index ( const std::string & name ) const** [inline], [inherited]

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.14.4.14 **StringVecT mappel::PointEmitterModel::get\_hyperparam\_names ( ) const** [inline],[inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.14.4.15 **double mappel::PointEmitterModel::get\_hyperparam\_value ( const std::string & name ) const** [inline], [inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

9.14.4.16 **PointEmitterModel::ParamT** mappel::PointEmitterModel::get\_hyperparams ( ) const [inline],  
[inherited]

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.14.4.17 **ImageFormat2DBase::ImageT** mappel::ImageFormat2DBase::get\_image\_from\_stack ( const ImageStackT &  
*stack*, ImageCoordT *n* ) const [inline],[inherited]

Definition at line 106 of file ImageFormat2DBase.h.

9.14.4.18 **const PointEmitterModel::ParamT &** mappel::PointEmitterModel::get\_lbound ( ) const [inline],  
[inherited]

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

9.14.4.19 **IdxT** mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases ( ) const [inherited]

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

9.14.4.20 **double** mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale ( ) const [inherited]

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

9.14.4.21 **IdxT** mappel::PointEmitterModel::get\_num\_hyperparams ( ) const [inline],[inherited]

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

9.14.4.22 **IdxT** mappel::PointEmitterModel::get\_num\_params ( ) const [inline],[inherited]

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.14.4.23 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get\_num\_pixels ( ) const** [inline], [inherited]

Definition at line 79 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

Referenced by mappel::ImageFormat2DBase::get\_stats().

**9.14.4.24 StringVecT mappel::PointEmitterModel::get\_param\_names ( ) const** [inline], [inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.14.4.25 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline], [inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_↔  
estimators().

**9.14.4.26 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline], [inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.14.4.27 const VecT & mappel::Gauss2DModel::get\_psf\_sigma ( ) const** [inline]

Definition at line 142 of file Gauss2DModel.h.

References psf\_sigma.

Referenced by get\_stats().

**9.14.4.28 double mappel::Gauss2DModel::get\_psf\_sigma ( IdxT idx ) const**

Definition at line 132 of file Gauss2DModel.cpp.

References DefaultPriorType, prior\_types, and psf\_sigma.

**9.14.4.29 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static], [inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.



#### 9.14.4.30 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( ) [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

#### 9.14.4.31 const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get\_size ( ) const [inline],[inherited]

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

#### 9.14.4.32 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get\_size ( IdxT idx ) const [inherited]

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

#### 9.14.4.33 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get\_size\_image\_stack ( const ImageStackT & stack ) const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

#### 9.14.4.34 StatsT mappel::Gauss2DModel::get\_stats ( ) const

Definition at line 268 of file Gauss2DModel.cpp.

References get\_psf\_sigma(), mappel::MCMCAdaptor2D::get\_stats(), mappel::ImageFormat2DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

#### 9.14.4.35 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

#### 9.14.4.36 bool mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

#### 9.14.4.37 **Gauss2DModel::Stencil** mappel::Gauss2DModel::initial\_theta\_estimate ( const ImageT & *im* ) const [inline]

Fast, heuristic estimate of initial theta.

Definition at line 194 of file Gauss2DModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::PointEmitterModel::make\_param().

Referenced by initial\_theta\_estimate().

#### 9.14.4.38 **Gauss2DModel::Stencil** mappel::Gauss2DModel::initial\_theta\_estimate ( const ImageT & *im*, const ParamT & *theta\_init* ) const [inline]

Definition at line 201 of file Gauss2DModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and initial\_theta\_estimate().

#### 9.14.4.39 **Gauss2DModel::Stencil** mappel::Gauss2DModel::initial\_theta\_estimate ( const ImageT & *im*, const ParamT & *theta\_init*, const std::string & *estimator* ) const

Definition at line 303 of file Gauss2DModel.cpp.

References mappel::Gauss2DModel::Stencil::bg(), mappel::methods::estimate\_max(), mappel::Gauss2DModel::Stencil::l(), mappel::PointEmitterModel::lbound, make\_stencil(), mappel::PointEmitterModel::num\_params, mappel::ImageFormat2DBase::size, mappel::estimator::MLEData::theta, mappel::PointEmitterModel::theta\_in\_bounds(), mappel::PointEmitterModel::ubound, x\_model, and y\_model.

#### 9.14.4.40 **CompositeDist** mappel::Gauss2DModel::make\_default\_prior ( const ImageSizeT & *size*, const std::string & *prior\_type* ) [static]

Definition at line 150 of file Gauss2DModel.cpp.

References mappel::istarts\_with(), make\_default\_prior\_beta\_position(), and make\_default\_prior\_normal\_position().

#### 9.14.4.41 **CompositeDist** mappel::Gauss2DModel::make\_default\_prior\_beta\_position ( const ImageSizeT & *size* ) [static]

Definition at line 171 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), and mappel::ImageFormat2DBase::size.

Referenced by make\_default\_prior().

#### 9.14.4.42 **CompositeDist** mappel::Gauss2DModel::make\_default\_prior\_normal\_position ( const ImageSizeT & *size* ) [static]

Definition at line 182 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), and mappel::ImageFormat2DBase::size.

Referenced by make\_default\_prior().

9.14.4.43 **ImageFormat2DBase::ImageT** mappel::ImageFormat2DBase::make\_image ( ) const [inline], [inherited]

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

9.14.4.44 **ImageFormat2DBase::ImageStackT** mappel::ImageFormat2DBase::make\_image\_stack ( ImageCoordT *n* ) const [inline], [inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

9.14.4.45 **Gauss2DModel::Gauss1DSumModelT** mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator ( IdxT *dim*, const ImageSizeT & *size*, const VecT & *psf\_sigma*, const CompositeDist & *prior* ) [static], [protected]

Definition at line 62 of file Gauss2DModel.cpp.

References mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), psf\_sigma, and mappel::ImageFormat2DBase::size.

Referenced by debug\_internal\_sum\_model\_y(), Gauss2DModel(), and update\_internal\_1Dsum\_estimators().

9.14.4.46 **PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline], [inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

9.14.4.47 **template<class FillT > PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( FillT *fill* ) const [inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.14.4.48 **MatT** mappel::PointEmitterModel::make\_param\_mat ( ) const [inline], [inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.14.4.49 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.14.4.50 `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.14.4.51 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`  
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.14.4.52 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta↔\_stack().

9.14.4.53 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.14.4.54 `CompositeDist mappel::Gauss2DModel::make_prior_beta_position ( const ImageSizeT & size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg )` `[static]`

Definition at line 193 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior↔\_component\_position\_beta(), and mappel::ImageFormat2DBase::size.

9.14.4.55 **prior\_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make\_prior\_component\_intensity ( double *mean* = default\_mean\_l, double *kappa* = default\_intensity\_kappa ) [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_l.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.14.4.56 **prior\_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make\_prior\_component\_position\_beta ( IdxT *size*, double *pos\_beta* = default\_beta\_pos ) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), and mappel::Gauss2DsModel::make\_prior\_beta\_position().

9.14.4.57 **prior\_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make\_prior\_component\_position\_normal ( IdxT *size*, double *pos\_sigma* = default\_sigma\_pos ) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.14.4.58 **prior\_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make\_prior\_component\_sigma ( double *min\_sigma*, double *max\_sigma*, double *alpha* = default\_alpha\_sigma ) [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.14.4.59 **CompositeDist** mappel::Gauss2DModel::make\_prior\_normal\_position ( const ImageSizeT & *size*, double *sigma\_xpos*, double *beta\_ypos*, double *mean\_l*, double *kappa\_l*, double *mean\_bg*, double *kappa\_bg* ) [static]

Definition at line 206 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), and mappel::ImageFormat2DBase::size.

**9.14.4.60 Gauss2DModel::Stencil mappel::Gauss2DModel::make\_stencil ( const ParamT & theta, bool compute\_derivatives = true ) const [inline]**

Make a new Model::Stencil object at theta.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular theta (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not model.theta\_in\_bounds(theta).

If derivatives will not be computed with this stencil set compute\_derivatives=false

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 131 of file Gauss2DModel.h.

References [mappel::Gauss2DModel::Stencil::Stencil\(\)](#), and [mappel::PointEmitterModel::theta\\_in\\_bounds\(\)](#).

Referenced by [initial\\_theta\\_estimate\(\)](#).

**9.14.4.61 Gauss2DModel & mappel::Gauss2DModel::operator= ( const Gauss2DModel & o ) [protected]**

Definition at line 38 of file Gauss2DModel.cpp.

References [mappel::MCMCAdaptor2D::operator=\(\)](#), [psf\\_sigma](#), [x\\_model](#), and [y\\_model](#).

Referenced by [debug\\_internal\\_sum\\_model\\_y\(\)](#), [mappel::Gauss2DModel::operator=\(\)](#), and [mappel::Gauss2DModel::operator=\(\)](#).

**9.14.4.62 Gauss2DModel & mappel::Gauss2DModel::operator= ( Gauss2DModel && o ) [protected]**

Definition at line 49 of file Gauss2DModel.cpp.

References [mappel::MCMCAdaptor2D::operator=\(\)](#), [psf\\_sigma](#), [x\\_model](#), and [y\\_model](#).

**9.14.4.63 void mappel::Gauss2DModel::pixel\_grad ( int i, int j, const Stencil & s, ParamT & pgrad ) const [inline]**

Definition at line 159 of file Gauss2DModel.h.

References [mappel::Gauss2DModel::Stencil::DX](#), [mappel::Gauss2DModel::Stencil::DY](#), [mappel::Gauss2DModel::Stencil::I\(\)](#), [mappel::Gauss2DModel::Stencil::X](#), and [mappel::Gauss2DModel::Stencil::Y](#).

Referenced by [pixel\\_hess\\_update\(\)](#).

9.14.4.64 `void mappel::Gauss2DModel::pixel_grad2 ( int i, int j, const Stencil & s, ParamT & pgrad2 ) const` `[inline]`

Definition at line 170 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

9.14.4.65 `void mappel::Gauss2DModel::pixel_hess ( int i, int j, const Stencil & s, MatT & hess ) const` `[inline]`

Definition at line 181 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

9.14.4.66 `void mappel::Gauss2DModel::pixel_hess_update ( int i, int j, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess ) const`

pixel derivative inner loop calculations.

Definition at line 282 of file Gauss2DModel.cpp.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::PointEmitterModel::make_param()`, `pixel_grad()`, `psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

9.14.4.67 `double mappel::Gauss2DModel::pixel_model_value ( int i, int j, const Stencil & s ) const` `[inline]`

Definition at line 152 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::bg()`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

9.14.4.68 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

9.14.4.69 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

9.14.4.70 `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline],[inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.14.4.71 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale`  
`=1.0 ) const [inherited]`

Definition at line 59 of file `MCMCAdaptor2D.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_↔`  
`x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.14.4.72 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT &`  
`fixed_parameters_mask, double step_scale = 1.0 ) const [inherited]`

Definition at line 74 of file `MCMCAdaptor2D.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_↔`  
`x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.14.4.73 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.14.4.74 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const [inline],[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.14.4.75 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 ) [inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::Point↔`  
`EmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_↔`  
`ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.



9.14.4.76 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.14.4.77 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline]`,  
`[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.14.4.78 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline]`,  
`[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.14.4.79 `void mappel::Gauss2DModel::set_hyperparams ( const VecT & hyperparams )`

Definition at line 109 of file `Gauss2DModel.cpp`.

References `mappel::PointEmitterModel::set_hyperparams()`, and `update_internal_1Dsum_estimators()`.

9.14.4.80 `template<class ImT > void mappel::ImageFormat2DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const` `[inherited]`

Definition at line 113 of file `ImageFormat2DBase.h`.

9.14.4.81 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 )` `[inherited]`

Definition at line 65 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.14.4.82 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.14.4.83 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` `[protected]`, `[inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.14.4.84 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` `[inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.14.4.85 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` `[inline]`, `[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.14.4.86 `void mappel::Gauss2DModel::set_prior ( CompositeDist && prior_ )`

Definition at line 97 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `update_internal_1Dsum_estimators()`.

9.14.4.87 `void mappel::Gauss2DModel::set_prior ( const CompositeDist & prior_ )`

Definition at line 103 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `update_internal_1Dsum_estimators()`.

9.14.4.88 `void mappel::Gauss2DModel::set_psf_sigma ( double new_psf_sigma )` `[inline]`

Definition at line 146 of file Gauss2DModel.h.

9.14.4.89 void mappel::Gauss2DModel::set\_psf\_sigma ( const VecT & new\_psf\_sigma )

Definition at line 123 of file Gauss2DModel.cpp.

References mappel::PointEmitterModel::check\_psf\_sigma(), psf\_sigma, mappel::Gauss1DModel::set\_psf\_sigma(), x\_model, and y\_model.

9.14.4.90 void mappel::PointEmitterModel::set\_rng\_seed ( RngSeedT seed ) [static], [inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.14.4.91 void mappel::Gauss2DModel::set\_size ( const ImageSizeT & size\_ )

Definition at line 115 of file Gauss2DModel.cpp.

References mappel::ImageFormat2DBase::set\_size(), mappel::ImageFormat1DBase::set\_size(), mappel::ImageFormat2DBase::size, x\_model, and y\_model.

9.14.4.92 void mappel::PointEmitterModel::set\_ubound ( const ParamT & ubound ) [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_max\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.14.4.93 bool mappel::PointEmitterModel::theta\_in\_bounds ( const ParamT & theta ) const [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::make\_stencil(), mappel::Gauss1DsModel::make\_stencil(), make\_stencil(), mappel::Gauss2DsxyModel::make\_stencil(), mappel::Gauss2DsModel::make\_stencil(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.14.4.94 BoolVecT mappel::PointEmitterModel::theta\_stack\_in\_bounds ( const ParamVecT & theta ) const [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::theta\_in\_bounds().

9.14.4.95 `void mappel::Gauss2DModel::update_internal_1Dsum_estimators ( )` [protected]

Definition at line 91 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::get_prior()`, `make_internal_1Dsum_estimator()`, `psf_sigma`, `mappel::Image2DFormat2DBase::size`, `x_model`, and `y_model`.

Referenced by `debug_internal_sum_model_y()`, `set_hyperparams()`, and `set_prior()`.

#### 9.14.5 Member Data Documentation

9.14.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static],[inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

9.14.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.14.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.14.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

9.14.5.5 `const double mappel::PointEmitterModel::default_max_I = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.14.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.14.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.14.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

9.14.5.9 `const std::string mappel::Gauss2DModel::DefaultPriorType = "Normal"` `[static]`

Definition at line 51 of file Gauss2DModel.h.

Referenced by `get_psf_sigma()`.

9.14.5.10 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.14.5.11 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

#### 9.14.5.12 `double mappel::MCMCAdaptor1D::eta_l=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

#### 9.14.5.13 `double mappel::MCMCAdaptor1D::eta_x=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

#### 9.14.5.14 `double mappel::MCMCAdaptor2D::eta_y=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor2D.h`.

Referenced by `mappel::MCMCAdaptor2D::get_stats()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2D::operator=()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`.

#### 9.14.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static],[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

#### 9.14.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static],[inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

#### 9.14.5.17 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static],[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.14.5.18** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` `[static]`,  
`[inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.14.5.19** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static]`, `[inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.14.5.20** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` `[static]`,  
`[inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.14.5.21** `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.14.5.22** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim = 2` `[static]`,  
`[inherited]`

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

**9.14.5.23** `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

#### 9.14.5.24 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.14.5.25 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

#### 9.14.5.26 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.14.5.27 `const StringVecT mappel::Gauss2DModel::prior_types` `[static]`

**Initial value:**

```
= { "Beta",
    "Normal"
}
```

Definition at line 50 of file `Gauss2DModel.h`.

Referenced by `get_psf_sigma()`.



#### 9.14.5.28 VecT mappel::Gauss2DModel::psf\_sigma [protected]

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 104 of file Gauss2DModel.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_← heuristic\_compute\_estimate(), mappel::Gauss2DModel::Stencil::compute\_derivatives(), debug\_internal\_sum\_model\_← \_y(), Gauss2DModel(), get\_psf\_sigma(), make\_internal\_1Dsum\_estimator(), operator=(), pixel\_grad2(), pixel\_hess(), pixel\_hess\_update(), set\_psf\_sigma(), mappel::Gauss2DModel::Stencil::Stencil(), and update\_internal\_1Dsum\_← estimators().

#### 9.14.5.29 double mappel::MCMCAdaptorBase::sigma\_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMC\_← Adaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set\_← \_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMC\_← AdaptorBase::set\_mcmc\_sigma\_scale().

#### 9.14.5.30 ImageSizeT mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_← \_heuristic\_compute\_estimate(), mappel::ImageFormat2DBase::check\_image\_shape(), mappel::Gauss2DModel::\_← Stencil::compute\_derivatives(), mappel::Gauss2DsModel::Stencil::compute\_derivatives(), debug\_internal\_sum\_← model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::methods::expected\_information(), Gauss2DModel(), mappel::Gauss2DsMAP::Gauss2DsMAP(), mappel::Gauss2DsMLE::Gauss2DsMLE(), mappel\_← ::Gauss2DsModel::Gauss2DsModel(), mappel::ImageFormat2DBase::get\_num\_pixels(), mappel::ImageFormat2D\_← Base::get\_size(), mappel::ImageFormat2DBase::get\_stats(), mappel::methods::likelihood::grad(), mappel::methods\_← ::likelihood::grad2(), mappel::methods::likelihood::debug::grad\_components(), mappel::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian\_components(), initial\_theta\_estimate(), mappel::Gauss2DsModel\_← ::initial\_theta\_estimate(), mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh\_components(), make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), make\_default\_← prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::ImageFormat2\_← DBase::make\_image(), mappel::ImageFormat2DBase::make\_image\_stack(), make\_internal\_1Dsum\_estimator(), mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator(), make\_prior\_beta\_position(), mappel::Gauss2Ds\_← Model::make\_prior\_beta\_position(), make\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_prior\_normal\_← \_position(), mappel::methods::model\_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods\_← ::likelihood::rllh(), mappel::methods::likelihood::debug::rllh\_components(), mappel::ImageFormat2DBase::set\_size(), set\_size(), mappel::Gauss2DsModel::set\_size(), mappel::methods::simulate\_image(), mappel::methods::simulate\_← image\_from\_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2DsModel::Stencil::Stencil(), update\_← \_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

#### 9.14.5.31 `ParamT mappel::PointEmitterModel::ubound` [protected], [inherited]

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.14.5.32 `Gauss1DSumModelT mappel::Gauss2DModel::x_model` [protected]

X-model fits 2D images X-axis (column sum)

Definition at line 105 of file `Gauss2DModel.h`.

Referenced by `debug_internal_sum_model_x()`, `initial_theta_estimate()`, `operator=()`, `set_psf_sigma()`, `set_size()`, and `update_internal_1Dsum_estimators()`.

#### 9.14.5.33 `Gauss1DSumModelT mappel::Gauss2DModel::y_model` [protected]

Y-model fits 2D images Y-axis (row sum)

Definition at line 106 of file `Gauss2DModel.h`.

Referenced by `debug_internal_sum_model_y()`, `Gauss2DModel()`, `initial_theta_estimate()`, `operator=()`, `set_psf_sigma()`, `set_size()`, and `update_internal_1Dsum_estimators()`.

The documentation for this class was generated from the following files:

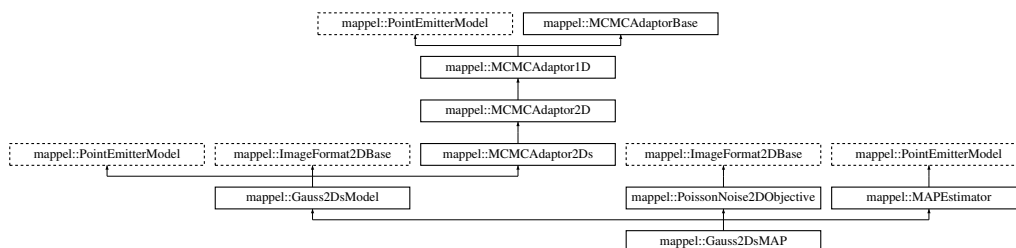
- [Gauss2DModel.h](#)
- [Gauss2DModel.cpp](#)

## 9.15 `mappel::Gauss2DsMAP` Class Reference

A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum a-posteriori objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DsMAP.h>
```

Inheritance diagram for `mappel::Gauss2DsMAP`:



## Public Types

- using [Gauss1DSumModelT](#) = [Gauss1DsMAP](#)
- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >
- using [ModelDataT](#) = [ImageT](#)
- using [ModelDataStackT](#) = [ImageStackT](#)

## Public Member Functions

- [Gauss2DsMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, double max\_sigma\_ratio, const std::string &prior\_type=DefaultPriorType)
- [Gauss2DsMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma, const std::string &prior\_type=DefaultPriorType)
- [Gauss2DsMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, CompositeDist &&prior)
- [Gauss2DsMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const CompositeDist &prior)
- [Gauss2DsMAP](#) (const [Gauss2DsMAP](#) &o)
- [Gauss2DsMAP](#) & operator= (const [Gauss2DsMAP](#) &o)
- [Gauss2DsMAP](#) ([Gauss2DsMAP](#) &&o)
- [Gauss2DsMAP](#) & operator= ([Gauss2DsMAP](#) &&o)
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- [VecT](#) [get\\_min\\_sigma](#) () const
- double [get\\_min\\_sigma](#) (IdxT dim) const
- [VecT](#) [get\\_max\\_sigma](#) () const
- double [get\\_max\\_sigma](#) (IdxT dim) const
- double [get\\_max\\_sigma\\_ratio](#) () const
- void [set\\_min\\_sigma](#) (const [VecT](#) &min\_sigma)
- void [set\\_max\\_sigma](#) (const [VecT](#) &max\_sigma)
- void [set\\_max\\_sigma\\_ratio](#) (double max\_sigma\_ratio)
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &theta, bool compute\_derivatives=true) const

*Make a new Model::Stencil object at theta.*

- double `pixel_model_value` (int i, int j, const `Stencil` &s) const
- void `pixel_grad` (int i, int j, const `Stencil` &s, `ParamT` &pgrad) const
- void `pixel_grad2` (int i, int j, const `Stencil` &s, `ParamT` &pgrad2) const
- void `pixel_hess` (int i, int j, const `Stencil` &s, `MatT` &hess) const
- void `pixel_hess_update` (int i, int j, const `Stencil` &s, double dm\_ratio\_m1, double dmm\_ratio, `ParamT` &grad, `MatT` &hess) const
- pixel derivative inner loop calculations.*
- `Stencil initial_theta_estimate` (const `ImageT` &im) const
  - Fast, heuristic estimate of initial theta.*
- `Stencil initial_theta_estimate` (const `ImageT` &im, const `ParamT` &theta\_init) const
- `Stencil initial_theta_estimate` (const `ImageT` &im, const `ParamT` &theta\_init, const std::string &estimator) const
- `Gauss1DSumModelT debug_internal_sum_model_x` () const
- `Gauss1DSumModelT debug_internal_sum_model_y` () const
- `IdxT get_num_params` () const
- void `check_param_shape` (const `ParamT` &theta) const
- void `check_param_shape` (const `ParamVecT` &theta) const
- void `check_psf_sigma` (double psf\_sigma) const
- void `check_psf_sigma` (const `VecT` &psf\_sigma) const
- `ParamT make_param` () const
- template<class FillT >
  - `ParamT make_param` (FillT fill) const
- `ParamVecT make_param_stack` (`IdxT` n) const
- template<class FillT >
  - `ParamVecT make_param_stack` (`IdxT` n, FillT fill) const
- `MatT make_param_mat` () const
- template<class FillT >
  - `MatT make_param_mat` (FillT fill) const
- `CubeT make_param_mat_stack` (`IdxT` n) const
- template<class FillT >
  - `CubeT make_param_mat_stack` (`IdxT` n, FillT fill) const
- `CompositeDist & get_prior` ()
- const `CompositeDist & get_prior` () const
- `IdxT get_num_hyperparams` () const
- `VecT get_hyperparams` () const
- bool `has_hyperparam` (const std::string &name) const
- double `get_hyperparam_value` (const std::string &name) const
- int `get_hyperparam_index` (const std::string &name) const
- void `set_hyperparam_value` (const std::string &name, double value)
- void `rename_hyperparam` (const std::string &old\_name, const std::string &new\_name)
- `StringVecT get_param_names` () const
- void `set_param_names` (const `StringVecT` &desc)
- `StringVecT get_hyperparam_names` () const
- void `set_hyperparam_names` (const `StringVecT` &desc)
- template<class RngT >
  - `ParamT sample_prior` (RngT &rng) const
- `ParamT sample_prior` () const
- void `set_bounds` (const `ParamT` &lbound, const `ParamT` &ubound)
- void `set_lbound` (const `ParamT` &lbound)
- void `set_ubound` (const `ParamT` &ubound)
- const `ParamT & get_lbound` () const
- const `ParamT & get_ubound` () const

- bool `theta_in_bounds` (const `ParamT` &theta) const
- void `bound_theta` (`ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `bounded_theta` (const `ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `reflected_theta` (const `ParamT` &theta) const
- `BoolVecT` `theta_stack_in_bounds` (const `ParamVecT` &theta) const
- `ParamVecT` `bounded_theta_stack` (const `ParamVecT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamVecT` `reflected_theta_stack` (const `ParamVecT` &theta) const
- `ImageT` `make_image` () const
- `ImageStackT` `make_image_stack` (`ImageCoordT` n) const
- `ImageCoordT` `get_size_image_stack` (const `ImageStackT` &stack) const
- `ImageT` `get_image_from_stack` (const `ImageStackT` &stack, `ImageCoordT` n) const
- template<class `ImT` >  
void `set_image_in_stack` (`ImageStackT` &stack, `ImageCoordT` n, const `ImT` &im) const
- const `ImageSizeT` & `get_size` () const
- `ImageCoordT` `get_size` (`IdxT` idx) const
- `ImageCoordT` `get_num_pixels` () const
- void `check_image_shape` (const `ImageT` &im) const  
*Check the shape of a single images is correct for model size.*
- void `check_image_shape` (const `ImageStackT` &ims) const  
*Check the shape of a stack of images is correct for model size.*
- void `sample_mcmc_candidate` (`IdxT` sample\_index, `ParamT` &candidate, double step\_scale=1.0) const
- void `sample_mcmc_candidate` (`IdxT` sample\_index, `ParamT` &candidate, const `IdxVecT` &fixed\_parameters\_↵  
mask, double step\_scale=1.0) const
- void `set_intensity_mcmc_sampling` (double eta\_l=-1)
- void `set_background_mcmc_sampling` (double eta\_bg=-1)
- void `set_mcmc_sigma_scale` (double scale)
- double `get_mcmc_sigma_scale` () const
- `IdxT` `get_mcmc_num_phases` () const

#### Static Public Member Functions

- static `CompositeDist` `make_default_prior` (const `ImageSizeT` &size, double max\_sigma\_ratio, const std::string &prior\_type)
- static `CompositeDist` `make_default_prior_beta_position` (const `ImageSizeT` &size, double max\_sigma\_ratio)
- static `CompositeDist` `make_default_prior_normal_position` (const `ImageSizeT` &size, double max\_sigma\_ratio)
- static `CompositeDist` `make_prior_beta_position` (const `ImageSizeT` &size, double beta\_xpos, double beta\_ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double max\_sigma\_ratio, double alpha\_↵  
sigma)
- static `CompositeDist` `make_prior_normal_position` (const `ImageSizeT` &size, double sigma\_xpos, double sigma\_↵  
\_ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double max\_sigma\_ratio, double alpha\_sigma)
- static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` size, double pos\_↵  
sigma=`default_sigma_pos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos\_↵  
beta=`default_beta_pos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=`default_mean_↵  
l`, double kappa=`default_intensity_kappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min\_sigma, double max\_↵  
sigma, double alpha=`default_alpha_sigma`)
- static void `set_rng_seed` (`RngSeedT` seed)

- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)

*Check the size argument for the model.*

#### Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) =2
- static const [ImageCoordT](#) [global\\_min\\_size](#) =3
- static const [ImageCoordT](#) [global\\_max\\_size](#) =512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5
- static const std::vector< std::string > [estimator\\_names](#)

#### Protected Member Functions

- void [update\\_internal\\_1Dsum\\_estimators](#) ()
- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) num\_phases)

#### Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make\\_internal\\_1Dsum\\_estimator](#) ([IdxT](#) dim, const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma, const [CompositeDist](#) &prior)
- static double [compute\\_max\\_sigma\\_ratio](#) (const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma)

## Protected Attributes

- [VecT min\\_sigma](#)
- [Gauss1DSumModelT x\\_model](#)
- [Gauss1DSumModelT y\\_model](#)
- [CompositeDist prior](#)
- [IdxT num\\_params](#)
- [IdxT num\\_hyperparams](#)
- [ParamT lbound](#)
- [ParamT ubound](#)
- [ImageSizeT size](#)
- [double eta\\_sigma](#) =0
- [double eta\\_y](#) =0
- [double eta\\_x](#) =0
- [double eta\\_l](#) =0
- [double eta\\_bg](#) =0
- [IdxT num\\_phases](#)
- [double sigma\\_scale](#)

## 9.15.1 Detailed Description

A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum a-posteriori objective.

Model: [Gauss2DsModel](#) - 2D Gaussian variable scalar PSF sigma Objective: [PoissonNoise2DObjective](#) - Poisson noise model for 2D Estimator: [MAPEstimator](#) - Maximum a-posteriori estimator

Definition at line 24 of file Gauss2DsMAP.h.

## 9.15.2 Member Typedef Documentation

9.15.2.1 `using mappel::Gauss2DsModel::Gauss1DSumModelT = Gauss1DsMAP` [\[inherited\]](#)

Definition at line 26 of file Gauss2DsModel.h.

9.15.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` [\[inherited\]](#)

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

9.15.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` [\[inherited\]](#)

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

**9.15.2.4** `template<class PixelT > using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>`  
[inherited]

Shape of the data type for a single image

Definition at line 32 of file ImageFormat2DBase.h.

**9.15.2.5** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>`  
[inherited]

Shape of the data type to store a single image's coordinates

Definition at line 27 of file ImageFormat2DBase.h.

**9.15.2.6** `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`  
[inherited]

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

**9.15.2.7** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>`  
[inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

**9.15.2.8** `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
[inherited]

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

**9.15.2.9** `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`  
[inherited]

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

**9.15.2.10** `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
[inherited]

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.



9.15.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT> [inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

9.15.2.12 `using mappel::PoissonNoise2DObjective::ModelDataStackT = ImageStackT [inherited]`

Objective function data stack type: 2D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise2DObjective.h.

9.15.2.13 `using mappel::PoissonNoise2DObjective::ModelDataT = ImageT [inherited]`

Objective function data type: 2D double precision image, gain-corrected to approximate photons counts

Definition at line 24 of file PoissonNoise2DObjective.h.

9.15.2.14 `using mappel::PointEmitterModel::ParamT = arma::vec [inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.15.2.15 `using mappel::PointEmitterModel::ParamVecT = arma::mat [inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

9.15.2.16 `using mappel::Gauss2DsModel::StencilVecT = std::vector<Stencil> [inherited]`

Definition at line 55 of file Gauss2DsModel.h.

### 9.15.3 Constructor & Destructor Documentation

9.15.3.1 `mappel::Gauss2DsMAP::Gauss2DsMAP ( const ImageSizeT & size, const VecT & min_sigma, double max_sigma_ratio, const std::string & prior_type = DefaultPriorType )`

Definition at line 11 of file Gauss2DsMAP.cpp.

References `mappel::Gauss2DsModel::min_sigma`, and `mappel::ImageFormat2DBase::size`.

9.15.3.2 `mappel::Gauss2DsMAP::Gauss2DsMAP ( const ImageSizeT & size, const VecT & min_sigma, const VecT & max_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 15 of file Gauss2DsMAP.cpp.

References `mappel::Gauss2DsModel::compute_max_sigma_ratio()`, `mappel::Gauss2DsModel::make_default_prior()`, `mappel::Gauss2DsModel::min_sigma`, and `mappel::ImageFormat2DBase::size`.

9.15.3.3 `mappel::Gauss2DsMAP::Gauss2DsMAP ( const ImageSizeT & size, const VecT & min_sigma, CompositeDist && prior )`

Definition at line 19 of file Gauss2DsMAP.cpp.

9.15.3.4 `mappel::Gauss2DsMAP::Gauss2DsMAP ( const ImageSizeT & size, const VecT & min_sigma, const CompositeDist & prior )`

Definition at line 27 of file Gauss2DsMAP.cpp.

9.15.3.5 `mappel::Gauss2DsMAP::Gauss2DsMAP ( const Gauss2DsMAP & o )`

Definition at line 35 of file Gauss2DsMAP.cpp.

9.15.3.6 `mappel::Gauss2DsMAP::Gauss2DsMAP ( Gauss2DsMAP && o )`

Definition at line 43 of file Gauss2DsMAP.cpp.

#### 9.15.4 Member Function Documentation

9.15.4.1 `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const`  
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

9.15.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

**9.15.4.3** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

**9.15.4.4** `void mappel::ImageFormat2DBase::check_image_shape ( const ImageT & im ) const` [inherited]

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

**9.15.4.5** `void mappel::ImageFormat2DBase::check_image_shape ( const ImageStackT & ims ) const` [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

**9.15.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.15.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

**9.15.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

9.15.4.9 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

9.15.4.10 `void mappel::ImageFormat2DBase::check_size ( const ImageSizeT & size_ )` [static],[inherited]

Check the size argument for the model.

Definition at line 60 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::global_max_size`, and `mappel::ImageFormat2DBase::global_min_size`.

Referenced by `mappel::ImageFormat2DBase::ImageFormat2DBase()`, and `mappel::ImageFormat2DBase::set_size()`.

9.15.4.11 `double mappel::Gauss2DsModel::compute_max_sigma_ratio ( const VecT & min_sigma, const VecT & max_sigma )` [static],[protected],[inherited]

Definition at line 162 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::min_sigma`.

Referenced by `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `Gauss2DsMAP()`, `mappel::Gauss2DsMLE::Gauss2DsMLE()`, and `mappel::Gauss2DsModel::set_max_sigma()`.

9.15.4.12 `Gauss1DSumModelT mappel::Gauss2DsModel::debug_internal_sum_model_x ( ) const` [inline],[inherited]

Definition at line 104 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::x_model`.

9.15.4.13 `Gauss1DSumModelT mappel::Gauss2DsModel::debug_internal_sum_model_y ( ) const` [inline],[inherited]

Definition at line 105 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::compute_max_sigma_ratio()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss2DsModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DsModel::min_sigma`, `mappel::Gauss2DsModel::operator=()`, `mappel::PointEmitterModel::prior`, `mappel::ImageFormat2DBase::size`, `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::y_model`.

9.15.4.14 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline],[inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.15.4.15 **StringVecT** mappel::PointEmitterModel::get\_hyperparam\_names ( ) const [inline], [inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.15.4.16 **double** mappel::PointEmitterModel::get\_hyperparam\_value ( const std::string & *name* ) const [inline], [inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

9.15.4.17 **PointEmitterModel::ParamT** mappel::PointEmitterModel::get\_hyperparams ( ) const [inline], [inherited]

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.15.4.18 **ImageFormat2DBase::ImageT** mappel::ImageFormat2DBase::get\_image\_from\_stack ( const ImageStackT & *stack*, ImageCoordT *n* ) const [inline], [inherited]

Definition at line 106 of file ImageFormat2DBase.h.

9.15.4.19 **const PointEmitterModel::ParamT &** mappel::PointEmitterModel::get\_lbound ( ) const [inline], [inherited]

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

9.15.4.20 **VecT** mappel::Gauss2DsModel::get\_max\_sigma ( ) const [inline], [inherited]

Definition at line 132 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), and mappel::Gauss2DsModel::get\_min\_sigma().

Referenced by mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss2DsModel::get\_stats(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), mappel::Gauss2DsModel::set\_min\_sigma(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

9.15.4.21 `double mappel::Gauss2DsModel::get_max_sigma ( IdxT dim ) const [inline],[inherited]`

Definition at line 136 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::get_max_sigma_ratio()`, and `mappel::Gauss2DsModel::get_min_sigma()`.

9.15.4.22 `double mappel::Gauss2DsModel::get_max_sigma_ratio ( ) const [inline],[inherited]`

Definition at line 140 of file Gauss2DsModel.h.

References `mappel::PointEmitterModel::get_ubound()`.

Referenced by `mappel::Gauss2DsModel::get_max_sigma()`, and `mappel::Gauss2DsModel::get_stats()`.

9.15.4.23 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const [inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

9.15.4.24 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const [inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

9.15.4.25 `VecT mappel::Gauss2DsModel::get_min_sigma ( ) const [inline],[inherited]`

Definition at line 128 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::min_sigma`.

Referenced by `mappel::Gauss2DsModel::get_max_sigma()`, `mappel::Gauss2DsModel::get_stats()`, and `mappel::Gauss2DsModel::set_max_sigma()`.

9.15.4.26 `double mappel::Gauss2DsModel::get_min_sigma ( IdxT dim ) const [inherited]`

Definition at line 191 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::DefaultPriorType`, `mappel::Gauss2DsModel::min_sigma`, and `mappel::Gauss2DsModel::prior_types`.

9.15.4.27 `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const [inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

**9.15.4.28** `IdxT mappel::PointEmitterModel::get_num_params ( ) const [inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.15.4.29** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_num_pixels ( ) const [inline],[inherited]`

Definition at line 79 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

Referenced by mappel::ImageFormat2DBase::get\_stats().

**9.15.4.30** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const [inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.15.4.31** `CompositeDist & mappel::PointEmitterModel::get_prior ( ) [inline],[inherited]`

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel↵  
::update\_internal\_1Dsum\_estimators().

**9.15.4.32** `const CompositeDist & mappel::PointEmitterModel::get_prior ( ) const [inline],[inherited]`

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.15.4.33** `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ( ) [static],[inherited]`

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.15.4.34** `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ( ) [static],[inherited]`

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.15.4.35** `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size ( ) const [inline], [inherited]`

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.15.4.36** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size ( IdxT idx ) const [inherited]`

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.15.4.37** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack ( const ImageStackT & stack ) const [inline], [inherited]`

Definition at line 99 of file ImageFormat2DBase.h.

**9.15.4.38** `StatsT mappel::Gauss2DsModel::get_stats ( ) const [inherited]`

Definition at line 337 of file Gauss2DsModel.cpp.

References mappel::Gauss2DsModel::get\_max\_sigma(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_min\_sigma(), mappel::MCMCAdaptor2Ds::get\_stats(), mappel::ImageFormat2DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

**9.15.4.39** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound ( ) const [inline], [inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

**9.15.4.40** `bool mappel::PointEmitterModel::has_hyperparam ( const std::string & name ) const [inline], [inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.15.4.41** `Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate ( const ImageT & im ) const [inline], [inherited]`

Fast, heuristic estimate of initial theta.

Definition at line 224 of file Gauss2DsModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::PointEmitterModel::make\_param().

Referenced by mappel::Gauss2DsModel::initial\_theta\_estimate().



**9.15.4.42** `Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init ) const` `[inline],[inherited]`

Definition at line 231 of file Gauss2DsModel.h.

References `mappel::PointEmitterModel::DefaultSeperableInitEstimator`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.15.4.43** `Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init, const std::string & estimator ) const` `[inherited]`

Definition at line 381 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::Stencil::bg()`, `mappel::methods::estimate_max()`, `mappel::Gauss2DsModel::Stencil::l()`, `mappel::PointEmitterModel::lbound`, `mappel::Gauss2DsModel::make_stencil()`, `mappel::Gauss2DsModel::min_sigma`, `mappel::PointEmitterModel::num_params`, `mappel::Gauss2DsModel::Stencil::sigma_ratio()`, `mappel::ImageFormat2DBase::size`, `mappel::estimator::MLEData::theta`, `mappel::PointEmitterModel::theta_in_bounds()`, `mappel::PointEmitterModel::ubound`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

**9.15.4.44** `CompositeDist mappel::Gauss2DsModel::make_default_prior ( const ImageSizeT & size, double max_sigma_ratio, const std::string & prior_type )` `[static],[inherited]`

Definition at line 208 of file Gauss2DsModel.cpp.

References `mappel::istarts_with()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, and `mappel::Gauss2DsModel::make_default_prior_normal_position()`.

Referenced by `Gauss2DsMAP()`, and `mappel::Gauss2DsMLE::Gauss2DsMLE()`.

**9.15.4.45** `CompositeDist mappel::Gauss2DsModel::make_default_prior_beta_position ( const ImageSizeT & size, double max_sigma_ratio )` `[static],[inherited]`

Definition at line 229 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::Gauss2DsModel::make_default_prior()`.

**9.15.4.46** `CompositeDist mappel::Gauss2DsModel::make_default_prior_normal_position ( const ImageSizeT & size, double max_sigma_ratio )` `[static],[inherited]`

Definition at line 241 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::Gauss2DsModel::make_default_prior()`.

**9.15.4.47 ImageFormat2DBase::ImageT** mappel::ImageFormat2DBase::make\_image ( ) const [inline],  
[inherited]

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.15.4.48 ImageFormat2DBase::ImageStackT** mappel::ImageFormat2DBase::make\_image\_stack ( ImageCoordT *n* ) const  
[inline], [inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.15.4.49 Gauss2DsModel::Gauss1DSumModelT** mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator ( IdxT *dim*,  
const ImageSizeT & *size*, const VecT & *min\_sigma*, const VecT & *max\_sigma*, const CompositeDist & *prior* )  
[static], [protected], [inherited]

Definition at line 69 of file Gauss2DsModel.cpp.

References mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_↵  
position(), mappel::Gauss2DsModel::min\_sigma, and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::Gauss2Ds↵  
Model(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.15.4.50 PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline], [inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(),  
mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::↵  
Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel↵  
::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_↵  
update().

**9.15.4.51 template<class FillT > PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( FillT *fill* ) const  
[inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.15.4.52 MatT** mappel::PointEmitterModel::make\_param\_mat ( ) const [inline], [inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.15.4.53 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.15.4.54 `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline], [inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.15.4.55 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`  
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.15.4.56 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline], [inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.15.4.57 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.15.4.58 `CompositeDist mappel::Gauss2DsModel::make_prior_beta_position ( const ImageSizeT & size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma )` `[static], [inherited]`

Definition at line 253 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), mappel::PointEmitterModel::make\_prior\_component\_sigma(), and mappel::ImageFormat2DBase::size.

9.15.4.59 **prior\_hessian::TruncatedGammaDist mappel::PointEmitterModel::make\_prior\_component\_intensity ( double mean = default\_mean\_I, double kappa = default\_intensity\_kappa )** [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_I.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.15.4.60 **prior\_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make\_prior\_component\_position\_beta ( IdxT size, double pos\_beta = default\_beta\_pos )** [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), and mappel::Gauss2DsModel::make\_prior\_beta\_position().

9.15.4.61 **prior\_hessian::TruncatedNormalDist mappel::PointEmitterModel::make\_prior\_component\_position\_normal ( IdxT size, double pos\_sigma = default\_sigma\_pos )** [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.15.4.62 **prior\_hessian::TruncatedParetoDist mappel::PointEmitterModel::make\_prior\_component\_sigma ( double min\_sigma, double max\_sigma, double alpha = default\_alpha\_sigma )** [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.15.4.63 **CompositeDist** mappel::Gauss2DsModel::make\_prior\_normal\_position ( const ImageSizeT & size, double sigma\_xpos, double sigma\_ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double max\_sigma\_ratio, double alpha\_sigma ) [static],[inherited]

Definition at line 268 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), mappel::PointEmitterModel::make\_prior\_component\_sigma(), and mappel::ImageFormat2DBase::size.

9.15.4.64 **Gauss2DsModel::Stencil** mappel::Gauss2DsModel::make\_stencil ( const ParamT & theta, bool compute\_derivatives =true ) const [inline],[inherited]

Make a new Model::Stencil object at theta.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular theta (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not model.theta\_in\_bounds(theta).

If derivatives will not be computed with this stencil set compute\_derivatives=false

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 162 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::Stencil(), and mappel::PointEmitterModel::theta\_in\_bounds().

Referenced by mappel::Gauss2DsModel::initial\_theta\_estimate().

9.15.4.65 **Gauss2DsMAP & mappel::Gauss2DsMAP::operator=** ( const Gauss2DsMAP & o )

Definition at line 51 of file Gauss2DsMAP.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::ImageFormat2DBase::operator=(), mappel::Gauss2DsModel::operator=(), and mappel::PointEmitterModel::operator=().

#### 9.15.4.66 Gauss2DsMAP & mappel::Gauss2DsMAP::operator= ( Gauss2DsMAP && o )

Definition at line 62 of file Gauss2DsMAP.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::Image↵  
Format2DBase::operator=(), mappel::Gauss2DsModel::operator=(), and mappel::PointEmitterModel::operator=().

#### 9.15.4.67 void mappel::Gauss2DsModel::pixel\_grad ( int i, int j, const Stencil & s, ParamT & pgrad ) const [inline], [inherited]

Definition at line 180 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵  
Model::Stencil::DY, mappel::Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::l(), mappel::Gauss2↵  
DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

Referenced by mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.15.4.68 void mappel::Gauss2DsModel::pixel\_grad2 ( int i, int j, const Stencil & s, ParamT & pgrad2 ) const [inline], [inherited]

Definition at line 192 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2DsModel::Stencil::DXS2, mappel::Gauss2↵  
DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::l(), mappel::↵  
Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), mappel::Gauss2DsModel::Stencil::X,  
and mappel::Gauss2DsModel::Stencil::Y.

#### 9.15.4.69 void mappel::Gauss2DsModel::pixel\_hess ( int i, int j, const Stencil & s, MatT & hess ) const [inline], [inherited]

Definition at line 204 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵  
Model::Stencil::DXS2, mappel::Gauss2DsModel::Stencil::DXSX, mappel::Gauss2DsModel::Stencil::DY, mappel::↵  
Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::DYSY,  
mappel::Gauss2DsModel::Stencil::l(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil↵  
::sigmaY(), mappel::Gauss2DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

#### 9.15.4.70 void mappel::Gauss2DsModel::pixel\_hess\_update ( int i, int j, const Stencil & s, double dm\_ratio\_m1, double dmm\_ratio, ParamT & grad, MatT & hess ) const [inherited]

pixel derivative inner loop calculations.

Definition at line 354 of file Gauss2DsModel.cpp.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵  
Model::Stencil::DXS2, mappel::Gauss2DsModel::Stencil::DXSX, mappel::Gauss2DsModel::Stencil::DY, mappel::↵  
Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::DYSY,  
mappel::Gauss2DsModel::Stencil::l(), mappel::PointEmitterModel::make\_param(), mappel::Gauss2DsModel::pixel\_↵  
grad(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), mappel::Gauss2Ds↵  
Model::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

**9.15.4.71** `double mappel::Gauss2DsModel::pixel_model_value ( int i, int j, const Stencil & s ) const` `[inline]`,  
`[inherited]`

Definition at line 173 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::Stencil::bg()`, `mappel::Gauss2DsModel::Stencil::l()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

**9.15.4.72** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.15.4.73** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.15.4.74** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.15.4.75** `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptor2Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor2Ds::eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.15.4.76** `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 75 of file MCMCAdaptor2Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor2Ds::eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.15.4.77 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.15.4.78 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline]`, `[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.15.4.79 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` `[inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.15.4.80 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.15.4.81 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline]`,  
`[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.15.4.82 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline]`,  
`[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.



9.15.4.83 void mappel::Gauss2DsModel::set\_hyperparams ( const VecT & *hyperparams* ) [inherited]

Definition at line 119 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::set\_hyperparams(), and mappel::Gauss2DsModel::update\_internal\_1Dsum←  
\_estimators().

9.15.4.84 template<class ImT > void mappel::ImageFormat2DBase::set\_image\_in\_stack ( ImageStackT & *stack*,  
ImageCoordT *n*, const ImT & *im* ) const [inherited]

Definition at line 113 of file ImageFormat2DBase.h.

9.15.4.85 void mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling ( double *eta\_l* = -1 ) [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_mean\_l, mappel::MCMCAdaptor1D::eta\_l, mappel::PointEmitter←  
Model::get\_hyperparam\_value(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

9.15.4.86 void mappel::PointEmitterModel::set\_lbound ( const ParamT & *lbound* ) [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitter←  
Model::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.15.4.87 void mappel::Gauss2DsModel::set\_max\_sigma ( const VecT & *new\_sigma* ) [inherited]

Set the max\_sigma\_ratio based on the new max\_sigma's ratio with the current min\_sigma.

Definition at line 155 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::check\_psf\_sigma(), mappel::Gauss2DsModel::compute\_max\_sigma\_ratio(),  
mappel::Gauss2DsModel::get\_min\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.15.4.88 void mappel::Gauss2DsModel::set\_max\_sigma\_ratio ( double *max\_sigma\_ratio* ) [inherited]

Definition at line 176 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::Gauss2DsModel::get\_max\_sigma(), mappel::Point←  
EmitterModel::get\_ubound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::PointEmitterModel::set\_ubound(),  
mappel::Gauss2DsModel::x\_model, and mappel::Gauss2DsModel::y\_model.

Referenced by mappel::Gauss2DsModel::set\_max\_sigma().

9.15.4.89 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` [protected],[inherited]

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.15.4.90 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` [inherited]

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.15.4.91 `void mappel::Gauss2DsModel::set_min_sigma ( const VecT & new_sigma )` [inherited]

Set the minimum sigma, keeping the `max_sigma_ratio` the same.

Definition at line 137 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `mappel::Gauss2DsModel::get_max_sigma()`, `mappel::Gauss2DsModel::min_sigma`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

9.15.4.92 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.15.4.93 `void mappel::Gauss2DsModel::set_prior ( CompositeDist && prior_ )` [inherited]

Definition at line 107 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

9.15.4.94 `void mappel::Gauss2DsModel::set_prior ( const CompositeDist & prior_ )` [inherited]

Definition at line 113 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

9.15.4.95 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` [static],[inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.15.4.96 `void mappel::Gauss2DsModel::set_size ( const ImageSizeT & size_ )` [inherited]

Definition at line 125 of file Gauss2DsModel.cpp.

References `mappel::ImageFormat2DBase::set_size()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::ImageFormat2DBase::size`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

9.15.4.97 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

9.15.4.98 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const` [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

9.15.4.99 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const` [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

9.15.4.100 `void mappel::Gauss2DsModel::update_internal_1Dsum_estimators ( )` [protected], [inherited]

Definition at line 100 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::get_max_sigma()`, `mappel::PointEmitterModel::get_prior()`, `mappel::Gauss2DsModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DsModel::min_sigma`, `mappel::ImageFormat2DBase::size`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

Referenced by `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_prior()`.

### 9.15.5 Member Data Documentation

**9.15.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

**9.15.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file `PointEmitterModel.h`.

**9.15.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file `PointEmitterModel.h`.

**9.15.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file `PointEmitterModel.h`.

**9.15.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.15.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file `PointEmitterModel.h`.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.15.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.15.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.15.5.9** `const std::string mappel::Gauss2DsModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 59 of file Gauss2DsModel.h.

Referenced by `mappel::Gauss2DsModel::get_min_sigma()`.

**9.15.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.15.5.11** `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` `[static], [inherited]`

Definition at line 23 of file PoissonNoise2DObjective.h.

**9.15.5.12** `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

#### 9.15.5.13 `double mappel::MCMCAdaptor1D::eta_l=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

#### 9.15.5.14 `double mappel::MCMCAdaptor2Ds::eta_sigma=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 27 of file `MCMCAdaptor2Ds.h`.

Referenced by `mappel::MCMCAdaptor2Ds::get_stats()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor2Ds::operator=()`, and `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`.

#### 9.15.5.15 `double mappel::MCMCAdaptor1D::eta_x=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

#### 9.15.5.16 `double mappel::MCMCAdaptor2D::eta_y=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor2D.h`.

Referenced by `mappel::MCMCAdaptor2D::get_stats()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2D::operator=()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`.

#### 9.15.5.17 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale=0.05` `[static]`, `[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.15.5.18 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

9.15.5.19 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.15.5.20 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

9.15.5.21 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.15.5.22 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

9.15.5.23 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.15.5.24 **VecT** mappel::Gauss2DsModel::min\_sigma [protected],[inherited]

Gaussian PSF in pixels

Definition at line 118 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::compute\_max\_sigma\_ratio(), mappel::Gauss2DsModel::debug\_internal←\_sum\_model\_y(), Gauss2DsMAP(), mappel::Gauss2DsMLE::Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2←DsModel(), mappel::Gauss2DsModel::get\_min\_sigma(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel←::Gauss2DsModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2Ds←Model::set\_min\_sigma(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

#### 9.15.5.25 **const std::string** mappel::Gauss2DsMAP::name [static]

Definition at line 36 of file Gauss2DsMAP.h.

#### 9.15.5.26 **const ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::num\_dim =2 [static], [inherited]

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::get\_stats().

#### 9.15.5.27 **IdxT** mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.15.5.28 **IdxT** mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel←::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitter←Model::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::Point←EmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitter←Model::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set←\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().



#### 9.15.5.29 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

#### 9.15.5.30 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.15.5.31 `const StringVecT mappel::Gauss2DsModel::prior_types` `[static]`, `[inherited]`

**Initial value:**

```
= { "Beta",
    "Normal"
}
```

Definition at line 58 of file Gauss2DsModel.h.

Referenced by `mappel::Gauss2DsModel::get_min_sigma()`.

#### 9.15.5.32 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`, `[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

### 9.15.5.33 **ImageSizeT** `mappel::ImageFormat2DBase::size` [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by `mappel::cgauss_compute_estimate()`, `mappel::cgauss_compute_estimate_debug()`, `mappel::cgauss_heuristic_compute_estimate()`, `mappel::ImageFormat2DBase::check_image_shape()`, `mappel::Gauss2DModel::Stencil::compute_derivatives()`, `mappel::Gauss2DsModel::Stencil::compute_derivatives()`, `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::methods::expected_information()`, `mappel::Gauss2DModel::Gauss2DModel()`, `Gauss2DsMAP()`, `mappel::Gauss2DsMLE::Gauss2DsMLE()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::ImageFormat2DBase::get_num_pixels()`, `mappel::ImageFormat2DBase::get_size()`, `mappel::ImageFormat2DBase::get_stats()`, `mappel::methods::likelihood::grad()`, `mappel::methods::likelihood::grad2()`, `mappel::methods::likelihood::debug::grad_components()`, `mappel::methods::likelihood::hessian()`, `mappel::methods::likelihood::debug::hessian_components()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::methods::likelihood::llh()`, `mappel::methods::likelihood::debug::llh_components()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::ImageFormat2DBase::make_image()`, `mappel::ImageFormat2DBase::make_image_stack()`, `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DsModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss2DsModel::make_prior_normal_position()`, `mappel::methods::model_image()`, `mappel::ImageFormat2DBase::operator=()`, `mappel::methods::likelihood::rllh()`, `mappel::methods::likelihood::debug::rllh_components()`, `mappel::ImageFormat2DBase::set_size()`, `mappel::Gauss2DModel::set_size()`, `mappel::Gauss2DsModel::set_size()`, `mappel::methods::simulate_image()`, `mappel::methods::simulate_image_from_model()`, `mappel::Gauss2DModel::Stencil::Stencil()`, `mappel::Gauss2DsModel::Stencil::Stencil()`, `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

### 9.15.5.34 **ParamT** `mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

### 9.15.5.35 **Gauss1DSumModelT** `mappel::Gauss2DsModel::x_model` [protected],[inherited]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 119 of file Gauss2DsModel.h.

Referenced by `mappel::Gauss2DsModel::debug_internal_sum_model_x()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::operator=()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_size()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

## 9.15.5.36 Gauss1DSumModelT mappel::Gauss2DsModel::y\_model [protected], [inherited]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 120 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_size(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

The documentation for this class was generated from the following files:

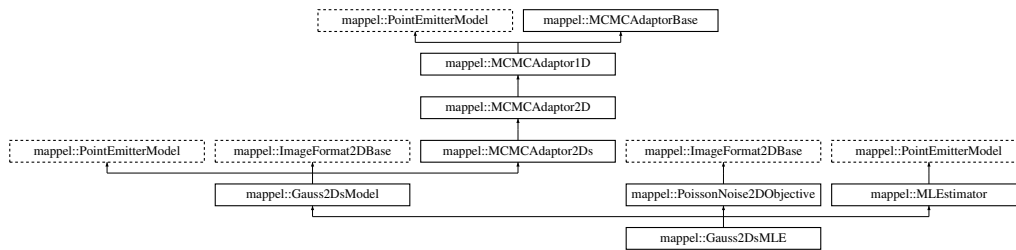
- [Gauss2DsMAP.h](#)
- [Gauss2DsMAP.cpp](#)

## 9.16 mappel::Gauss2DsMLE Class Reference

A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum-likelihood objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DsMLE.h>
```

Inheritance diagram for mappel::Gauss2DsMLE:



## Public Types

- using [Gauss1DSumModelT](#) = [Gauss1DsMAP](#)
- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >
- using [ModelDataT](#) = [ImageT](#)
- using [ModelDataStackT](#) = [ImageStackT](#)

## Public Member Functions

- [Gauss2DsMLE](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, double max\_sigma\_ratio, const std::string &prior\_type=DefaultPriorType)
- [Gauss2DsMLE](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma, const std::string &prior\_type=DefaultPriorType)
- [Gauss2DsMLE](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, CompositeDist &&prior)
- [Gauss2DsMLE](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const CompositeDist &prior)
- [Gauss2DsMLE](#) (const [Gauss2DsMLE](#) &o)
- [Gauss2DsMLE](#) & operator= (const [Gauss2DsMLE](#) &o)
- [Gauss2DsMLE](#) ([Gauss2DsMLE](#) &&o)
- [Gauss2DsMLE](#) & operator= ([Gauss2DsMLE](#) &&o)
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- [VecT](#) [get\\_min\\_sigma](#) () const
- double [get\\_min\\_sigma](#) (IdxT dim) const
- [VecT](#) [get\\_max\\_sigma](#) () const
- double [get\\_max\\_sigma](#) (IdxT dim) const
- double [get\\_max\\_sigma\\_ratio](#) () const
- void [set\\_min\\_sigma](#) (const [VecT](#) &min\_sigma)
- void [set\\_max\\_sigma](#) (const [VecT](#) &max\_sigma)
- void [set\\_max\\_sigma\\_ratio](#) (double max\_sigma\_ratio)
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &theta, bool compute\_derivatives=true) const  
*Make a new Model::Stencil object at theta.*
- double [pixel\\_model\\_value](#) (int i, int j, const [Stencil](#) &s) const
- void [pixel\\_grad](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel\\_grad2](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel\\_hess](#) (int i, int j, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel\\_hess\\_update](#) (int i, int j, const [Stencil](#) &s, double dm\_ratio\_m1, double dmm\_ratio, [ParamT](#) &grad, [MatT](#) &hess) const  
*pixel derivative inner loop calculations.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im) const  
*Fast, heuristic estimate of initial theta.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init) const
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init, const std::string &estimator) const
- [Gauss1DSumModelT](#) [debug\\_internal\\_sum\\_model\\_x](#) () const
- [Gauss1DSumModelT](#) [debug\\_internal\\_sum\\_model\\_y](#) () const
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT](#) [make\\_param](#) () const
- template<class FillT >  
[ParamT](#) [make\\_param](#) (FillT fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
[ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const

- [MatT make\\_param\\_mat](#) () const
- `template<class FillT >`  
[MatT make\\_param\\_mat](#) (FillT fill) const
- [CubeT make\\_param\\_mat\\_stack](#) (IdxT n) const
- `template<class FillT >`  
[CubeT make\\_param\\_mat\\_stack](#) (IdxT n, FillT fill) const
- [CompositeDist & get\\_prior](#) ()
- const [CompositeDist & get\\_prior](#) () const
- [IdxT get\\_num\\_hyperparams](#) () const
- [VecT get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- `template<class RngT >`  
[ParamT sample\\_prior](#) (RngT &rng) const
- [ParamT sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get\\_lbound](#) () const
- const [ParamT](#) & [get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT make\\_image](#) () const
- [ImageStackT make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- `template<class ImT >`  
void [set\\_image\\_in\\_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const ImT &im) const
- const [ImageSizeT](#) & [get\\_size](#) () const
- [ImageCoordT get\\_size](#) (IdxT idx) const
- [ImageCoordT get\\_num\\_pixels](#) () const
- void [check\\_image\\_shape](#) (const [ImageT](#) &im) const  
*Check the shape of a single images is correct for model size.*
- void [check\\_image\\_shape](#) (const [ImageStackT](#) &ims) const  
*Check the shape of a stack of images is correct for model size.*
- void [sample\\_mcmc\\_candidate](#) (IdxT sample\_index, [ParamT](#) &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) (IdxT sample\_index, [ParamT](#) &candidate, const [IdxVecT](#) &fixed\_parameters\_←  
mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)

- void [set\\_background\\_mcmc\\_sampling](#) (double [eta\\_bg](#)=-1)
- void [set\\_mcmc\\_sigma\\_scale](#) (double [scale](#))
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- [IdxT](#) [get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Member Functions

- static CompositeDist [make\\_default\\_prior](#) (const [ImageSizeT](#) &[size](#), double [max\\_sigma\\_ratio](#), const std::string &[prior\\_type](#))
- static CompositeDist [make\\_default\\_prior\\_beta\\_position](#) (const [ImageSizeT](#) &[size](#), double [max\\_sigma\\_ratio](#))
- static CompositeDist [make\\_default\\_prior\\_normal\\_position](#) (const [ImageSizeT](#) &[size](#), double [max\\_sigma\\_ratio](#))
- static CompositeDist [make\\_prior\\_beta\\_position](#) (const [ImageSizeT](#) &[size](#), double [beta\\_xpos](#), double [beta\\_ypos](#), double [mean\\_l](#), double [kappa\\_l](#), double [mean\\_bg](#), double [kappa\\_bg](#), double [max\\_sigma\\_ratio](#), double [alpha\\_sigma](#))
- static CompositeDist [make\\_prior\\_normal\\_position](#) (const [ImageSizeT](#) &[size](#), double [sigma\\_xpos](#), double [sigma\\_ypos](#), double [mean\\_l](#), double [kappa\\_l](#), double [mean\\_bg](#), double [kappa\\_bg](#), double [max\\_sigma\\_ratio](#), double [alpha\\_sigma](#))
- static [prior\\_hessian::TruncatedNormalDist](#) [make\\_prior\\_component\\_position\\_normal](#) ([IdxT](#) [size](#), double [pos\\_sigma](#)=[default\\_sigma\\_pos](#))
- static [prior\\_hessian::ScaledSymmetricBetaDist](#) [make\\_prior\\_component\\_position\\_beta](#) ([IdxT](#) [size](#), double [pos\\_beta](#)=[default\\_beta\\_pos](#))
- static [prior\\_hessian::TruncatedGammaDist](#) [make\\_prior\\_component\\_intensity](#) (double [mean](#)=[default\\_mean\\_l](#), double [kappa](#)=[default\\_intensity\\_kappa](#))
- static [prior\\_hessian::TruncatedParetoDist](#) [make\\_prior\\_component\\_sigma](#) (double [min\\_sigma](#), double [max\\_sigma](#), double [alpha](#)=[default\\_alpha\\_sigma](#))
- static void [set\\_rng\\_seed](#) ([RngSeedT](#) [seed](#))
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const [ImageSizeT](#) &[size](#))

*Check the size argument for the model.*

#### Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) =2
- static const [ImageCoordT](#) [global\\_min\\_size](#) =3
- static const [ImageCoordT](#) [global\\_max\\_size](#) =512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5
- static const std::vector< std::string > [estimator\\_names](#)

## Protected Member Functions

- void [update\\_internal\\_1Dsum\\_estimators](#) ()
- void [set\\_mcmc\\_num\\_phases](#) (IdxT num\_phases)

## Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make\\_internal\\_1Dsum\\_estimator](#) (IdxT dim, const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma, const [CompositeDist](#) &prior)
- static double [compute\\_max\\_sigma\\_ratio](#) (const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma)

## Protected Attributes

- [VecT](#) min\_sigma
- [Gauss1DSumModelT](#) x\_model
- [Gauss1DSumModelT](#) y\_model
- [CompositeDist](#) prior
- [IdxT](#) num\_params
- [IdxT](#) num\_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double eta\_sigma =0
- double eta\_y =0
- double eta\_x =0
- double eta\_l =0
- double eta\_bg =0
- [IdxT](#) num\_phases
- double sigma\_scale

## 9.16.1 Detailed Description

A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum-likelihood objective.

Model: [Gauss2DsModel](#) - 2D Gaussian variable scalar PSF sigma Objective: [PoissonNoise2DObjective](#) - Poisson noise model for 2D Estimator: [MLEstimator](#) - Pure-likelihood estimator

Definition at line 24 of file Gauss2DsMLE.h.

## 9.16.2 Member Typedef Documentation

## 9.16.2.1 using mappel::Gauss2DsModel::Gauss1DSumModelT = Gauss1DsMAP [inherited]

Definition at line 26 of file Gauss2DsModel.h.

**9.16.2.2** `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` `[inherited]`

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

**9.16.2.3** `using mappel::ImageFormat2DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

**9.16.2.4** `template<class PixelT > using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>`  
`[inherited]`

Shape of the data type for a single image

Definition at line 32 of file ImageFormat2DBase.h.

**9.16.2.5** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>`  
`[inherited]`

Shape of the data type to store a single image's coordinates

Definition at line 27 of file ImageFormat2DBase.h.

**9.16.2.6** `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`  
`[inherited]`

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

**9.16.2.7** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>`  
`[inherited]`

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

**9.16.2.8** `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.



**9.16.2.9** `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

**9.16.2.10** `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

**9.16.2.11** `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

**9.16.2.12** `using mappel::PoissonNoise2DObjective::ModelDataStackT = ImageStackT` `[inherited]`

Objective function data stack type: 2D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise2DObjective.h.

**9.16.2.13** `using mappel::PoissonNoise2DObjective::ModelDataT = ImageT` `[inherited]`

Objective function data type: 2D double precision image, gain-corrected to approximate photons counts

Definition at line 24 of file PoissonNoise2DObjective.h.

**9.16.2.14** `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

**9.16.2.15** `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

**9.16.2.16** `using mappel::Gauss2DsModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 55 of file Gauss2DsModel.h.

### 9.16.3 Constructor & Destructor Documentation

**9.16.3.1** `mappel::Gauss2DsMLE::Gauss2DsMLE ( const ImageSizeT & size, const VecT & min_sigma, double max_sigma_ratio, const std::string & prior_type = DefaultPriorType )`

Definition at line 11 of file Gauss2DsMLE.cpp.

References `mappel::Gauss2DsModel::min_sigma`, and `mappel::ImageFormat2DBase::size`.

**9.16.3.2** `mappel::Gauss2DsMLE::Gauss2DsMLE ( const ImageSizeT & size, const VecT & min_sigma, const VecT & max_sigma, const std::string & prior_type = DefaultPriorType )`

Definition at line 15 of file Gauss2DsMLE.cpp.

References `mappel::Gauss2DsModel::compute_max_sigma_ratio()`, `mappel::Gauss2DsModel::make_default_prior()`, `mappel::Gauss2DsModel::min_sigma`, and `mappel::ImageFormat2DBase::size`.

**9.16.3.3** `mappel::Gauss2DsMLE::Gauss2DsMLE ( const ImageSizeT & size, const VecT & min_sigma, CompositeDist && prior )`

Definition at line 19 of file Gauss2DsMLE.cpp.

**9.16.3.4** `mappel::Gauss2DsMLE::Gauss2DsMLE ( const ImageSizeT & size, const VecT & min_sigma, const CompositeDist & prior )`

Definition at line 27 of file Gauss2DsMLE.cpp.

**9.16.3.5** `mappel::Gauss2DsMLE::Gauss2DsMLE ( const Gauss2DsMLE & o )`

Definition at line 35 of file Gauss2DsMLE.cpp.

**9.16.3.6** `mappel::Gauss2DsMLE::Gauss2DsMLE ( Gauss2DsMLE && o )`

Definition at line 43 of file Gauss2DsMLE.cpp.

### 9.16.4 Member Function Documentation

**9.16.4.1** `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const`  
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

**9.16.4.2** `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

**9.16.4.3** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

**9.16.4.4** `void mappel::ImageFormat2DBase::check_image_shape ( const ImageT & im ) const` [inherited]

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

**9.16.4.5** `void mappel::ImageFormat2DBase::check_image_shape ( const ImageStackT & ims ) const` [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

**9.16.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.16.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

9.16.4.8 `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` `[inherited]`

Definition at line 185 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

9.16.4.9 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` `[inherited]`

Definition at line 197 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

9.16.4.10 `void mappel::ImageFormat2DBase::check_size ( const ImageSizeT & size_ )` `[static],[inherited]`

Check the size argument for the model.

Definition at line 60 of file `ImageFormat2DBase.cpp`.

References `mappel::ImageFormat2DBase::global_max_size`, and `mappel::ImageFormat2DBase::global_min_size`.

Referenced by `mappel::ImageFormat2DBase::ImageFormat2DBase()`, and `mappel::ImageFormat2DBase::set_size()`.

9.16.4.11 `double mappel::Gauss2DsModel::compute_max_sigma_ratio ( const VecT & min_sigma, const VecT & max_sigma )` `[static],[protected],[inherited]`

Definition at line 162 of file `Gauss2DsModel.cpp`.

References `mappel::Gauss2DsModel::min_sigma`.

Referenced by `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsMAP::Gauss2DsMAP()`, `Gauss2DsMLE()`, and `mappel::Gauss2DsModel::set_max_sigma()`.

9.16.4.12 `Gauss1DSumModelT mappel::Gauss2DsModel::debug_internal_sum_model_x ( ) const` `[inline],[inherited]`

Definition at line 104 of file `Gauss2DsModel.h`.

References `mappel::Gauss2DsModel::x_model`.

**9.16.4.13** `Gauss1DSumModelT` `mappel::Gauss2DsModel::debug_internal_sum_model_y ( ) const` `[inline]`,  
`[inherited]`

Definition at line 105 of file `Gauss2DsModel.h`.

References `mappel::Gauss2DsModel::compute_max_sigma_ratio()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss2DsModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DsModel::min_sigma`, `mappel::Gauss2DsModel::operator=()`, `mappel::PointEmitterModel::prior`, `mappel::ImageFormat2DBase::size`, `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::y_model`.

**9.16.4.14** `int` `mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 236 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.16.4.15** `StringVecT` `mappel::PointEmitterModel::get_hyperparam_names ( ) const` `[inline]`, `[inherited]`

Definition at line 256 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.16.4.16** `double` `mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 232 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.16.4.17** `PointEmitterModel::ParamT` `mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.16.4.18** `ImageFormat2DBase::ImageT` `mappel::ImageFormat2DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` `[inline]`, `[inherited]`

Definition at line 106 of file `ImageFormat2DBase.h`.

**9.16.4.19** `const PointEmitterModel::ParamT &` `mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.16.4.20 **VecT** mappel::Gauss2DsModel::get\_max\_sigma ( ) const [inline],[inherited]

Definition at line 132 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), and mappel::Gauss2DsModel::get\_min\_sigma().

Referenced by mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss2DsModel::get\_stats(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), mappel::Gauss2DsModel::set\_min\_sigma(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

9.16.4.21 **double** mappel::Gauss2DsModel::get\_max\_sigma ( IdxT dim ) const [inline],[inherited]

Definition at line 136 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), and mappel::Gauss2DsModel::get\_min\_sigma().

9.16.4.22 **double** mappel::Gauss2DsModel::get\_max\_sigma\_ratio ( ) const [inline],[inherited]

Definition at line 140 of file Gauss2DsModel.h.

References mappel::PointEmitterModel::get\_ubound().

Referenced by mappel::Gauss2DsModel::get\_max\_sigma(), and mappel::Gauss2DsModel::get\_stats().

9.16.4.23 **IdxT** mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases ( ) const [inherited]

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

9.16.4.24 **double** mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale ( ) const [inherited]

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

9.16.4.25 **VecT** mappel::Gauss2DsModel::get\_min\_sigma ( ) const [inline],[inherited]

Definition at line 128 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::min\_sigma.

Referenced by mappel::Gauss2DsModel::get\_max\_sigma(), mappel::Gauss2DsModel::get\_stats(), and mappel::Gauss2DsModel::set\_max\_sigma().

9.16.4.26 **double** mappel::Gauss2DsModel::get\_min\_sigma ( IdxT dim ) const [inherited]

Definition at line 191 of file Gauss2DsModel.cpp.

References mappel::Gauss2DsModel::DefaultPriorType, mappel::Gauss2DsModel::min\_sigma, and mappel::Gauss2DsModel::prior\_types.

**9.16.4.27** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline],[inherited]`

Definition at line 208 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_hyperparams`.

**9.16.4.28** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline],[inherited]`

Definition at line 160 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.16.4.29** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_num_pixels ( ) const` `[inline],[inherited]`

Definition at line 79 of file `ImageFormat2DBase.h`.

References `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

**9.16.4.30** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const` `[inline],[inherited]`

Definition at line 248 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.16.4.31** `CompositeDist & mappel::PointEmitterModel::get_prior ( )` `[inline],[inherited]`

Definition at line 200 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

**9.16.4.32** `const CompositeDist & mappel::PointEmitterModel::get_prior ( ) const` `[inline],[inherited]`

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.16.4.33** `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ( )` `[static],[inherited]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

**9.16.4.34 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( )** [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.16.4.35 const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get\_size ( )** const [inline],[inherited]

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.16.4.36 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get\_size ( IdxT idx )** const [inherited]

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.16.4.37 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get\_size\_image\_stack ( const ImageStackT & stack )** const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

**9.16.4.38 StatsT mappel::Gauss2DsModel::get\_stats ( )** const [inherited]

Definition at line 337 of file Gauss2DsModel.cpp.

References mappel::Gauss2DsModel::get\_max\_sigma(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_min\_sigma(), mappel::MCMCAdaptor2Ds::get\_stats(), mappel::ImageFormat2DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

**9.16.4.39 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( )** const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

**9.16.4.40 bool mappel::PointEmitterModel::has\_hyperparam ( const std::string & name )** const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.



**9.16.4.41 Gauss2DsModel::Stencil** mappel::Gauss2DsModel::initial\_theta\_estimate ( const ImageT & *im* ) const  
[inline], [inherited]

Fast, heuristic estimate of initial theta.

Definition at line 224 of file Gauss2DsModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::PointEmitterModel::make\_↵  
param().

Referenced by mappel::Gauss2DsModel::initial\_theta\_estimate().

**9.16.4.42 Gauss2DsModel::Stencil** mappel::Gauss2DsModel::initial\_theta\_estimate ( const ImageT & *im*, const ParamT &  
*theta\_init* ) const [inline], [inherited]

Definition at line 231 of file Gauss2DsModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::Gauss2DsModel::initial\_theta\_↵  
estimate().

**9.16.4.43 Gauss2DsModel::Stencil** mappel::Gauss2DsModel::initial\_theta\_estimate ( const ImageT & *im*, const ParamT &  
*theta\_init*, const std::string & *estimator* ) const [inherited]

Definition at line 381 of file Gauss2DsModel.cpp.

References mappel::Gauss2DsModel::Stencil::bg(), mappel::methods::estimate\_max(), mappel::Gauss2DsModel::↵  
Stencil::l(), mappel::PointEmitterModel::lbound, mappel::Gauss2DsModel::make\_stencil(), mappel::Gauss2DsModel↵  
::min\_sigma, mappel::PointEmitterModel::num\_params, mappel::Gauss2DsModel::Stencil::sigma\_ratio(), mappel↵  
::ImageFormat2DBase::size, mappel::estimator::MLEData::theta, mappel::PointEmitterModel::theta\_in\_bounds(),  
mappel::PointEmitterModel::ubound, mappel::Gauss2DsModel::x\_model, and mappel::Gauss2DsModel::y\_model.

**9.16.4.44 CompositeDist** mappel::Gauss2DsModel::make\_default\_prior ( const ImageSizeT & *size*, double *max\_sigma\_ratio*,  
const std::string & *prior\_type* ) [static], [inherited]

Definition at line 208 of file Gauss2DsModel.cpp.

References mappel::istarts\_with(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), and mappel::↵  
Gauss2DsModel::make\_default\_prior\_normal\_position().

Referenced by mappel::Gauss2DsMAP::Gauss2DsMAP(), and Gauss2DsMLE().

**9.16.4.45 CompositeDist** mappel::Gauss2DsModel::make\_default\_prior\_beta\_position ( const ImageSizeT & *size*, double  
*max\_sigma\_ratio* ) [static], [inherited]

Definition at line 229 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component↵  
\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), mappel::PointEmitterModel::make↵  
\_prior\_component\_sigma(), and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DsModel::make\_default\_prior().

**9.16.4.46 CompositeDist mappel::Gauss2DsModel::make\_default\_prior\_normal\_position ( const ImageSizeT & size, double max\_sigma\_ratio ) [static],[inherited]**

Definition at line 241 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component←\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_normal(), mappel::PointEmitterModel←::make\_prior\_component\_sigma(), and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DsModel::make\_default\_prior().

**9.16.4.47 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make\_image ( ) const [inline],[inherited]**

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.16.4.48 ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make\_image\_stack ( ImageCoordT n ) const [inline],[inherited]**

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.16.4.49 Gauss2DsModel::Gauss1DSumModelT mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator ( IdxT dim, const ImageSizeT & size, const VecT & min\_sigma, const VecT & max\_sigma, const CompositeDist & prior ) [static],[protected],[inherited]**

Definition at line 69 of file Gauss2DsModel.cpp.

References mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_normal←position(), mappel::Gauss2DsModel::min\_sigma, and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::Gauss2Ds←Model(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.16.4.50 PointEmitterModel::ParamT mappel::PointEmitterModel::make\_param ( ) const [inline],[inherited]**

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::←Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel←::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess←update().

9.16.4.51 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const`  
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.16.4.52 `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.16.4.53 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.16.4.54 `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.16.4.55 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`  
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.16.4.56 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.16.4.57 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.16.4.58 **CompositeDist** `mappel::Gauss2DsModel::make_prior_beta_position ( const ImageSizeT & size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma ) [static],[inherited]`

Definition at line 253 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

9.16.4.59 **prior\_hessian::TruncatedGammaDist** `mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_l, double kappa = default_intensity_kappa ) [static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.16.4.60 **prior\_hessian::ScaledSymmetricBetaDist** `mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos ) [static],[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

9.16.4.61 **prior\_hessian::TruncatedNormalDist** `mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos ) [static],[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.16.4.62 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma ) [static], [inherited]`

Definition at line 104 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.16.4.63 `CompositeDist mappel::Gauss2DsModel::make_prior_normal_position ( const ImageSizeT & size, double sigma_xpos, double sigma_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma ) [static], [inherited]`

Definition at line 268 of file `Gauss2DsModel.cpp`.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

9.16.4.64 `Gauss2DsModel::Stencil mappel::Gauss2DsModel::make_stencil ( const ParamT & theta, bool compute_derivatives = true ) const [inline], [inherited]`

Make a new `Model::Stencil` object at `theta`.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular `theta` (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 162 of file `Gauss2DsModel.h`.

References `mappel::Gauss2DsModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

Referenced by `mappel::Gauss2DsModel::initial_theta_estimate()`.

#### 9.16.4.65 **Gauss2DsMLE & mappel::Gauss2DsMLE::operator= ( const Gauss2DsMLE & o )**

Definition at line 51 of file Gauss2DsMLE.cpp.

References mappel::MLEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::Image↵  
Format2DBase::operator=(), mappel::Gauss2DsModel::operator=(), and mappel::PointEmitterModel::operator=().

#### 9.16.4.66 **Gauss2DsMLE & mappel::Gauss2DsMLE::operator= ( Gauss2DsMLE && o )**

Definition at line 62 of file Gauss2DsMLE.cpp.

References mappel::MLEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::Image↵  
Format2DBase::operator=(), mappel::Gauss2DsModel::operator=(), and mappel::PointEmitterModel::operator=().

#### 9.16.4.67 **void mappel::Gauss2DsModel::pixel\_grad ( int i, int j, const Stencil & s, ParamT & pgrad ) const [inline], [inherited]**

Definition at line 180 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵  
Model::Stencil::DY, mappel::Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::l(), mappel::Gauss2↵  
DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

Referenced by mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.16.4.68 **void mappel::Gauss2DsModel::pixel\_grad2 ( int i, int j, const Stencil & s, ParamT & pgrad2 ) const [inline], [inherited]**

Definition at line 192 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2DsModel::Stencil::DXS2, mappel::Gauss2↵  
DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::l(), mappel::↵  
Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), mappel::Gauss2DsModel::Stencil::X,  
and mappel::Gauss2DsModel::Stencil::Y.

#### 9.16.4.69 **void mappel::Gauss2DsModel::pixel\_hess ( int i, int j, const Stencil & s, MatT & hess ) const [inline], [inherited]**

Definition at line 204 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵  
Model::Stencil::DXS2, mappel::Gauss2DsModel::Stencil::DXSX, mappel::Gauss2DsModel::Stencil::DY, mappel::↵  
Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::DYSY,  
mappel::Gauss2DsModel::Stencil::l(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil↵  
::sigmaY(), mappel::Gauss2DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

9.16.4.70 `void mappel::Gauss2DsModel::pixel_hess_update ( int i, int j, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess ) const` [inherited]

pixel derivative inner loop calculations.

Definition at line 354 of file Gauss2DsModel.cpp.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2DsModel::Stencil::DXS2, mappel::Gauss2DsModel::Stencil::DXSX, mappel::Gauss2DsModel::Stencil::DY, mappel::Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::DYSY, mappel::Gauss2DsModel::Stencil::I(), mappel::PointEmitterModel::make\_param(), mappel::Gauss2DsModel::pixel\_grad(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), mappel::Gauss2DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

9.16.4.71 `double mappel::Gauss2DsModel::pixel_model_value ( int i, int j, const Stencil & s ) const` [inline], [inherited]

Definition at line 173 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::bg(), mappel::Gauss2DsModel::Stencil::I(), mappel::Gauss2DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

9.16.4.72 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const` [inherited]

Definition at line 276 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::reflected\_theta\_stack().

9.16.4.73 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` [inherited]

Definition at line 316 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::make\_param\_stack(), and mappel::PointEmitterModel::reflected\_theta().

9.16.4.74 `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )` [inline], [inherited]

Definition at line 244 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.16.4.75 `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const` [inherited]

Definition at line 56 of file MCMCAdaptor2Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor2Ds::eta_`  
`_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_`  
`phases`, and `mappel::rng_manager`.

9.16.4.76 `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const` [inherited]

Definition at line 75 of file MCMCAdaptor2Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor2Ds::eta_`  
`_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_`  
`phases`, and `mappel::rng_manager`.

9.16.4.77 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const` [inherited]

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.16.4.78 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` [inline], [inherited]

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.16.4.79 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::Point`  
`EmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_`  
`ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.16.4.80 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitter`  
`Model::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.



9.16.4.81 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc ) [inline], [inherited]`

Definition at line 260 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.16.4.82 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.16.4.83 `void mappel::Gauss2DsModel::set_hyperparams ( const VecT & hyperparams ) [inherited]`

Definition at line 119 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::set\_hyperparams(), and mappel::Gauss2DsModel::update\_internal\_1Dsum←\_estimators().

9.16.4.84 `template<class ImT > void mappel::ImageFormat2DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]`

Definition at line 113 of file ImageFormat2DBase.h.

9.16.4.85 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 ) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_mean\_I, mappel::MCMCAdaptor1D::eta\_I, mappel::PointEmitter←Model::get\_hyperparam\_value(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

9.16.4.86 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitter←Model::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.16.4.87 `void mappel::Gauss2DsModel::set_max_sigma ( const VecT & new_sigma ) [inherited]`

Set the max\_sigma\_ratio based on the new max\_sigma's ratio with the current min\_sigma.

Definition at line 155 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::check\_psf\_sigma(), mappel::Gauss2DsModel::compute\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_min\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.16.4.88 `void mappel::Gauss2DsModel::set_max_sigma_ratio ( double max_sigma_ratio )` `[inherited]`

Definition at line 176 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::Gauss2DsModel::get_max_sigma()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::PointEmitterModel::set_ubound()`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

Referenced by `mappel::Gauss2DsModel::set_max_sigma()`.

9.16.4.89 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` `[protected],[inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.16.4.90 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` `[inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.16.4.91 `void mappel::Gauss2DsModel::set_min_sigma ( const VecT & new_sigma )` `[inherited]`

Set the minimum sigma, keeping the `max_sigma_ratio` the same.

Definition at line 137 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `mappel::Gauss2DsModel::get_max_sigma()`, `mappel::Gauss2DsModel::min_sigma`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

9.16.4.92 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` `[inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.16.4.93 `void mappel::Gauss2DsModel::set_prior ( CompositeDist && prior )` `[inherited]`

Definition at line 107 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

9.16.4.94 `void mappel::Gauss2DsModel::set_prior ( const CompositeDist & prior_ ) [inherited]`

Definition at line 113 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_← estimators()`.

9.16.4.95 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.16.4.96 `void mappel::Gauss2DsModel::set_size ( const ImageSizeT & size_ ) [inherited]`

Definition at line 125 of file Gauss2DsModel.cpp.

References `mappel::ImageFormat2DBase::set_size()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Image← Format2DBase::size`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

9.16.4.97 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitter← Model::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

9.16.4.98 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::Point← EmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make← _stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::Point← EmitterModel::theta_stack_in_bounds()`.

9.16.4.99 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const [inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.16.4.100** `void mappel::Gauss2DsModel::update_internal_1Dsum_estimators ( )` [protected],[inherited]

Definition at line 100 of file Gauss2DsModel.cpp.

References mappel::Gauss2DsModel::get\_max\_sigma(), mappel::PointEmitterModel::get\_prior(), mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DsModel::min\_sigma, mappel::ImageFormat2DBase::size, mappel::Gauss2DsModel::x\_model, and mappel::Gauss2DsModel::y\_model.

Referenced by mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::set\_hyperparams(), and mappel::Gauss2DsModel::set\_prior().

## 9.16.5 Member Data Documentation

**9.16.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static],[inherited]

Distance from the boundary to constrain in bound\_theta and bounded\_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), and mappel::PointEmitterModel::set\_ubound().

**9.16.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

**9.16.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

**9.16.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.16.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::make\_prior\_component\_intensity().

9.16.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.16.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.16.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

9.16.5.9 `const std::string mappel::Gauss2DsModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 59 of file Gauss2DsModel.h.

Referenced by `mappel::Gauss2DsModel::get_min_sigma()`.

9.16.5.10 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.16.5.11 `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` `[static], [inherited]`

Definition at line 23 of file PoissonNoise2DObjective.h.

#### 9.16.5.12 `double mappel::MCMCAdaptor1D::eta_bg=0` [protected],[inherited]

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

#### 9.16.5.13 `double mappel::MCMCAdaptor1D::eta_l=0` [protected],[inherited]

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

#### 9.16.5.14 `double mappel::MCMCAdaptor2Ds::eta_sigma=0` [protected],[inherited]

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 27 of file `MCMCAdaptor2Ds.h`.

Referenced by `mappel::MCMCAdaptor2Ds::get_stats()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor2Ds::operator=()`, and `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`.

#### 9.16.5.15 `double mappel::MCMCAdaptor1D::eta_x=0` [protected],[inherited]

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

#### 9.16.5.16 `double mappel::MCMCAdaptor2D::eta_y=0` [protected],[inherited]

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor2D.h`.

Referenced by `mappel::MCMCAdaptor2D::get_stats()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2D::operator=()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`.

**9.16.5.17** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.16.5.18** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.16.5.19** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.16.5.20** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.16.5.21** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.16.5.22** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

#### 9.16.5.23 **ParamT** mappel::PointEmitterModel::lbound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.16.5.24 **VecT** mappel::Gauss2DsModel::min\_sigma [protected],[inherited]

Gaussian PSF in pixels

Definition at line 118 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::compute\_max\_sigma\_ratio(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsMAP::Gauss2DsMAP(), Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss2DsModel::get\_min\_sigma(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

#### 9.16.5.25 **const std::string** mappel::Gauss2DsMLE::name [static]

Definition at line 36 of file Gauss2DsMLE.h.

#### 9.16.5.26 **const ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::num\_dim =2 [static],[inherited]

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::get\_stats().

#### 9.16.5.27 **IdxT** mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.16.5.28 **IdxT** mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().



#### 9.16.5.29 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

#### 9.16.5.30 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.16.5.31 `const StringVecT mappel::Gauss2DsModel::prior_types` `[static]`, `[inherited]`

**Initial value:**

```
= { "Beta",
    "Normal"
}
```

Definition at line 58 of file Gauss2DsModel.h.

Referenced by `mappel::Gauss2DsModel::get_min_sigma()`.

#### 9.16.5.32 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`, `[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

### 9.16.5.33 **ImageSizeT** mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_←  
\_heuristic\_compute\_estimate(), mappel::ImageFormat2DBase::check\_image\_shape(), mappel::Gauss2DModel::←  
Stencil::compute\_derivatives(), mappel::Gauss2DsModel::Stencil::compute\_derivatives(), mappel::Gauss2DModel::←  
::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::methods::←  
::expected\_information(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsMAP::Gauss2DsMAP(),  
Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::ImageFormat2DBase::get\_num\_pixels(),  
mappel::ImageFormat2DBase::get\_size(), mappel::ImageFormat2DBase::get\_stats(), mappel::methods::likelihood←  
::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood::debug::grad\_components(), mappel←  
::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian\_components(), mappel::Gauss2D←  
Model::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::methods::likelihood←  
::llh(), mappel::methods::likelihood::debug::llh\_components(), mappel::Gauss2DModel::make\_default\_prior\_beta\_←  
position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_←  
prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::ImageFormat2D←  
Base::make\_image(), mappel::ImageFormat2DBase::make\_image\_stack(), mappel::Gauss2DModel::make\_internal←  
\_1Dsum\_estimator(), mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DModel::make\_←  
\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_←  
prior\_normal\_position(), mappel::Gauss2DsModel::make\_prior\_normal\_position(), mappel::methods::model\_image(),  
mappel::ImageFormat2DBase::operator=(), mappel::methods::likelihood::rllh(), mappel::methods::likelihood::debug←  
::rllh\_components(), mappel::ImageFormat2DBase::set\_size(), mappel::Gauss2DModel::set\_size(), mappel::Gauss2←  
DsModel::set\_size(), mappel::methods::simulate\_image(), mappel::methods::simulate\_image\_from\_model(), mappel←  
::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2DsModel::Stencil::Stencil(), mappel::Gauss2DModel::update\_←  
internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

### 9.16.5.34 **ParamT** mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel←  
::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta←  
\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(),  
mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel←  
::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::Point←  
EmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

### 9.16.5.35 **Gauss1DSumModelT** mappel::Gauss2DsModel::x\_model [protected],[inherited]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 119 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::debug\_internal\_sum\_model\_x(), mappel::Gauss2DsModel::initial\_theta←  
\_estimate(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), mappel::←  
Gauss2DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_size(), and mappel::Gauss2DsModel::update\_←  
internal\_1Dsum\_estimators().

#### 9.16.5.36 Gauss1DSumModelT mappel::Gauss2DsModel::y\_model [protected], [inherited]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 120 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_size(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

The documentation for this class was generated from the following files:

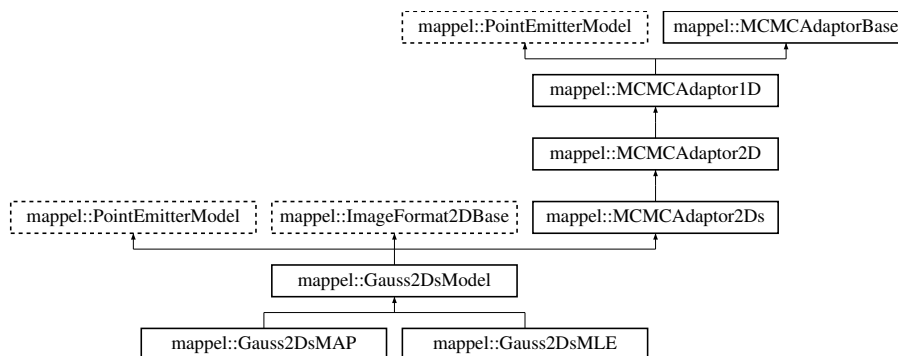
- [Gauss2DsMLE.h](#)
- [Gauss2DsMLE.cpp](#)

## 9.17 mappel::Gauss2DsModel Class Reference

A base class for 2D Gaussian PSF where the gaussian sigma is controlled by a single scalar parameter which is called sigma\_ratio. The size of the gaussian psf is sigma\_ratio\*psf\_sigma, where psf\_sigma is considered as a vector [psf\_sigmaX, psf\_sigmaY].

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DsModel.h>
```

Inheritance diagram for mappel::Gauss2DsModel:



### Classes

- class [Stencil](#)  
*Stencil for 2D scalar-sigma models.*

## Public Types

- using [Gauss1DSumModelT](#) = [Gauss1DsMAP](#)
- using [StencilVecT](#) = `std::vector< Stencil >`
- using [ParamT](#) = `arma::vec`
- using [ParamVecT](#) = `arma::mat`
- using [ImageCoordT](#) = `uint32_t`
- using [ImagePixelT](#) = `double`
- template<class [CoordT](#) >  
using [ImageSizeShapeT](#) = `arma::Col< CoordT >`
- template<class [CoordT](#) >  
using [ImageSizeVecShapeT](#) = `arma::Mat< CoordT >`
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class [PixelT](#) >  
using [ImageShapeT](#) = `arma::Mat< PixelT >`
- template<class [PixelT](#) >  
using [ImageStackShapeT](#) = `arma::Cube< PixelT >`
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >

## Public Member Functions

- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- [VecT](#) [get\\_min\\_sigma](#) () const
- double [get\\_min\\_sigma](#) ([IdxT](#) dim) const
- [VecT](#) [get\\_max\\_sigma](#) () const
- double [get\\_max\\_sigma](#) ([IdxT](#) dim) const
- double [get\\_max\\_sigma\\_ratio](#) () const
- void [set\\_min\\_sigma](#) (const [VecT](#) &min\_sigma)
- void [set\\_max\\_sigma](#) (const [VecT](#) &max\_sigma)
- void [set\\_max\\_sigma\\_ratio](#) (double max\_sigma\_ratio)
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &theta, bool compute\_derivatives=true) const  
*Make a new Model::Stencil object at theta.*
- double [pixel\\_model\\_value](#) (int i, int j, const [Stencil](#) &s) const
- void [pixel\\_grad](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel\\_grad2](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel\\_hess](#) (int i, int j, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel\\_hess\\_update](#) (int i, int j, const [Stencil](#) &s, double dm\_ratio\_m1, double dmm\_ratio, [ParamT](#) &grad, [MatT](#) &hess) const  
*pixel derivative inner loop calculations.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im) const  
*Fast, heuristic estimate of initial theta.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init) const
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init, const std::string &estimator) const
- [Gauss1DSumModelT](#) [debug\\_internal\\_sum\\_model\\_x](#) () const

- [Gauss1DSumModelT debug\\_internal\\_sum\\_model\\_y \(\)](#) const
- [IdxT get\\_num\\_params \(\)](#) const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT make\\_param \(\)](#) const
- template<class FillT >  
  [ParamT make\\_param](#) (FillT fill) const
- [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT make\\_param\\_mat \(\)](#) const
- template<class FillT >  
  [MatT make\\_param\\_mat](#) (FillT fill) const
- [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get\\_prior \(\)](#)
- const CompositeDist & [get\\_prior \(\)](#) const
- [IdxT get\\_num\\_hyperparams \(\)](#) const
- [VecT get\\_hyperparams \(\)](#) const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT get\\_param\\_names \(\)](#) const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT get\\_hyperparam\\_names \(\)](#) const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT sample\\_prior](#) (RngT &rng) const
- [ParamT sample\\_prior \(\)](#) const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get\\_lbound \(\)](#) const
- const [ParamT](#) & [get\\_ubound \(\)](#) const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT make\\_image \(\)](#) const
- [ImageStackT make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const

- `template<class ImT >`  
`void set_image_in_stack (ImageStackT &stack, ImageCoordT n, const ImT &im) const`
- `const ImageSizeT & get_size () const`
- `ImageCoordT get_size (IdxT idx) const`
- `ImageCoordT get_num_pixels () const`
- `void check_image_shape (const ImageT &im) const`  
*Check the shape of a single images is correct for model size.*
- `void check_image_shape (const ImageStackT &ims) const`  
*Check the shape of a stack of images is correct for model size.*
- `void sample_mcmc_candidate (IdxT sample_index, ParamT &candidate, double step_scale=1.0) const`
- `void sample_mcmc_candidate (IdxT sample_index, ParamT &candidate, const IdxVecT &fixed_parameters_↵  
mask, double step_scale=1.0) const`
- `void set_intensity_mcmc_sampling (double eta_l=-1)`
- `void set_background_mcmc_sampling (double eta_bg=-1)`
- `void set_mcmc_sigma_scale (double scale)`
- `double get_mcmc_sigma_scale () const`
- `IdxT get_mcmc_num_phases () const`

#### Static Public Member Functions

- static CompositeDist `make_default_prior` (const ImageSizeT &size, double max\_sigma\_ratio, const std::string &prior\_type)
- static CompositeDist `make_default_prior_beta_position` (const ImageSizeT &size, double max\_sigma\_ratio)
- static CompositeDist `make_default_prior_normal_position` (const ImageSizeT &size, double max\_sigma\_ratio)
- static CompositeDist `make_prior_beta_position` (const ImageSizeT &size, double beta\_xpos, double beta\_ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double max\_sigma\_ratio, double alpha\_↵  
sigma)
- static CompositeDist `make_prior_normal_position` (const ImageSizeT &size, double sigma\_xpos, double sigma\_↵  
\_ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double max\_sigma\_ratio, double alpha\_sigma)
- static prior\_hessian::TruncatedNormalDist `make_prior_component_position_normal` (IdxT size, double pos\_↵  
sigma=default\_sigma\_pos)
- static prior\_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta` (IdxT size, double pos\_↵  
beta=default\_beta\_pos)
- static prior\_hessian::TruncatedGammaDist `make_prior_component_intensity` (double mean=default\_mean\_↵  
l, double kappa=default\_intensity\_kappa)
- static prior\_hessian::TruncatedParetoDist `make_prior_component_sigma` (double min\_sigma, double max\_↵  
sigma, double alpha=default\_alpha\_sigma)
- static void `set_rng_seed` (RngSeedT seed)
- static ParallelRngManagerT & `get_rng_manager` ()
- static ParallelRngGeneratorT & `get_rng_generator` ()
- static void `check_size` (const ImageSizeT &size\_)

*Check the size argument for the model.*

## Static Public Attributes

- static const [StringVecT](#) [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) =2
- static const [ImageCoordT](#) [global\\_min\\_size](#) =3
- static const [ImageCoordT](#) [global\\_max\\_size](#) =512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5

## Protected Member Functions

- [Gauss2DsModel](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma)
- [Gauss2DsModel](#) (const [Gauss2DsModel](#) &o)
- [Gauss2DsModel](#) ([Gauss2DsModel](#) &&o)
- [Gauss2DsModel](#) & operator= (const [Gauss2DsModel](#) &o)
- [Gauss2DsModel](#) & operator= ([Gauss2DsModel](#) &&o)
- void [update\\_internal\\_1Dsum\\_estimators](#) ()
- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) num\_phases)

## Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make\\_internal\\_1Dsum\\_estimator](#) ([IdxT](#) dim, const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma, const [CompositeDist](#) &prior)
- static double [compute\\_max\\_sigma\\_ratio](#) (const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma)

## Protected Attributes

- [VecT](#) [min\\_sigma](#)
- [Gauss1DSumModelT](#) [x\\_model](#)
- [Gauss1DSumModelT](#) [y\\_model](#)
- [CompositeDist](#) [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta\\_sigma](#) =0
- double [eta\\_y](#) =0
- double [eta\\_x](#) =0
- double [eta\\_l](#) =0
- double [eta\\_bg](#) =0
- [IdxT](#) [num\\_phases](#)
- double [sigma\\_scale](#)

### 9.17.1 Detailed Description

A base class for 2D Gaussian PSF where the gaussian sigma is controlled by a single scalar parameter which is called `sigma_ratio`. The size of the gaussian psf is `sigma_ratio*psf_sigma`, where `psf_sigma` is considered as a vector [`psf_sigmaX`, `psf_sigmaY`].

Definition at line 23 of file `Gauss2DsModel.h`.

### 9.17.2 Member Typedef Documentation

#### 9.17.2.1 `using mappel::Gauss2DsModel::Gauss1DSumModelT = Gauss1DsMAP`

Definition at line 26 of file `Gauss2DsModel.h`.

#### 9.17.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 24 of file `ImageFormat2DBase.h`.

#### 9.17.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 25 of file `ImageFormat2DBase.h`.

#### 9.17.2.4 `template<class PixelT> using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>` [inherited]

Shape of the data type for a single image

Definition at line 32 of file `ImageFormat2DBase.h`.

#### 9.17.2.5 `template<class CoordT> using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>` [inherited]

Shape of the data type to store a single image's coordinates

Definition at line 27 of file `ImageFormat2DBase.h`.

#### 9.17.2.6 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 29 of file `ImageFormat2DBase.h`.



**9.17.2.7** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>`  
`[inherited]`

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

**9.17.2.8** `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

**9.17.2.9** `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

**9.17.2.10** `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

**9.17.2.11** `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

**9.17.2.12** `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

**9.17.2.13** `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

**9.17.2.14** `using mappel::Gauss2DsModel::StencilVecT = std::vector<Stencil>`

Definition at line 55 of file Gauss2DsModel.h.

### 9.17.3 Constructor & Destructor Documentation

**9.17.3.1** `mappel::Gauss2DsModel::Gauss2DsModel ( const ImageSizeT & size, const VecT & min_sigma, const VecT & max_sigma )` `[protected]`

Definition at line 12 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `make_internal_1Dsum_estimator()`, `min_sigma`, `mappel::PointEmitterModel::prior`, `mappel::ImageFormat2DBase::size`, and `y_model`.

Referenced by `debug_internal_sum_model_y()`.

**9.17.3.2** `mappel::Gauss2DsModel::Gauss2DsModel ( const Gauss2DsModel & o )` `[protected]`

Definition at line 28 of file Gauss2DsModel.cpp.

References `get_max_sigma()`, `make_internal_1Dsum_estimator()`, `min_sigma`, `mappel::PointEmitterModel::prior`, `mappel::ImageFormat2DBase::size`, and `y_model`.

**9.17.3.3** `mappel::Gauss2DsModel::Gauss2DsModel ( Gauss2DsModel && o )` `[protected]`

Definition at line 36 of file Gauss2DsModel.cpp.

References `get_max_sigma()`, `make_internal_1Dsum_estimator()`, `min_sigma`, `mappel::PointEmitterModel::prior`, `mappel::ImageFormat2DBase::size`, and `y_model`.

### 9.17.4 Member Function Documentation

**9.17.4.1** `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const` `[inherited]`

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

**9.17.4.2** `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` `[inherited]`

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

**9.17.4.3** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

**9.17.4.4** `void mappel::ImageFormat2DBase::check_image_shape ( const ImageT & im ) const` [inherited]

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

**9.17.4.5** `void mappel::ImageFormat2DBase::check_image_shape ( const ImageStackT & ims ) const` [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

**9.17.4.6** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.17.4.7** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

**9.17.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

9.17.4.9 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

9.17.4.10 `void mappel::ImageFormat2DBase::check_size ( const ImageSizeT & size_ )` `[static],[inherited]`

Check the size argument for the model.

Definition at line 60 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::global_max_size`, and `mappel::ImageFormat2DBase::global_min_size`.

Referenced by `mappel::ImageFormat2DBase::ImageFormat2DBase()`, and `mappel::ImageFormat2DBase::set_size()`.

9.17.4.11 `double mappel::Gauss2DsModel::compute_max_sigma_ratio ( const VecT & min_sigma, const VecT & max_sigma )` `[static],[protected]`

Definition at line 162 of file Gauss2DsModel.cpp.

References `min_sigma`.

Referenced by `debug_internal_sum_model_y()`, `mappel::Gauss2DsMAP::Gauss2DsMAP()`, `mappel::Gauss2DsMLE::Gauss2DsMLE()`, and `set_max_sigma()`.

9.17.4.12 `Gauss1DSumModelT mappel::Gauss2DsModel::debug_internal_sum_model_x ( ) const` `[inline]`

Definition at line 104 of file Gauss2DsModel.h.

References `x_model`.

9.17.4.13 `Gauss1DSumModelT mappel::Gauss2DsModel::debug_internal_sum_model_y ( ) const` `[inline]`

Definition at line 105 of file Gauss2DsModel.h.

References `compute_max_sigma_ratio()`, `Gauss2DsModel()`, `make_internal_1Dsum_estimator()`, `min_sigma`, `operator=()`, `mappel::PointEmitterModel::prior`, `mappel::ImageFormat2DBase::size`, `update_internal_1Dsum_estimators()`, and `y_model`.

9.17.4.14 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline],[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.17.4.15 **StringVecT** mappel::PointEmitterModel::get\_hyperparam\_names ( ) const [inline], [inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.17.4.16 **double** mappel::PointEmitterModel::get\_hyperparam\_value ( const std::string & name ) const [inline], [inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

9.17.4.17 **PointEmitterModel::ParamT** mappel::PointEmitterModel::get\_hyperparams ( ) const [inline], [inherited]

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.17.4.18 **ImageFormat2DBase::ImageT** mappel::ImageFormat2DBase::get\_image\_from\_stack ( const ImageStackT & stack, ImageCoordT n ) const [inline], [inherited]

Definition at line 106 of file ImageFormat2DBase.h.

9.17.4.19 **const PointEmitterModel::ParamT &** mappel::PointEmitterModel::get\_lbound ( ) const [inline], [inherited]

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

9.17.4.20 **VecT** mappel::Gauss2DsModel::get\_max\_sigma ( ) const [inline]

Definition at line 132 of file Gauss2DsModel.h.

References get\_max\_sigma\_ratio(), and get\_min\_sigma().

Referenced by Gauss2DsModel(), get\_stats(), operator=(), set\_max\_sigma\_ratio(), set\_min\_sigma(), and update\_internal\_1Dsum\_estimators().

**9.17.4.21** `double mappel::Gauss2DsModel::get_max_sigma ( IdxT dim ) const` `[inline]`

Definition at line 136 of file Gauss2DsModel.h.

References `get_max_sigma_ratio()`, and `get_min_sigma()`.

**9.17.4.22** `double mappel::Gauss2DsModel::get_max_sigma_ratio ( ) const` `[inline]`

Definition at line 140 of file Gauss2DsModel.h.

References `mappel::PointEmitterModel::get_ubound()`.

Referenced by `get_max_sigma()`, and `get_stats()`.

**9.17.4.23** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

**9.17.4.24** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

**9.17.4.25** `VecT mappel::Gauss2DsModel::get_min_sigma ( ) const` `[inline]`

Definition at line 128 of file Gauss2DsModel.h.

References `min_sigma`.

Referenced by `get_max_sigma()`, `get_stats()`, and `set_max_sigma()`.

**9.17.4.26** `double mappel::Gauss2DsModel::get_min_sigma ( IdxT dim ) const`

Definition at line 191 of file Gauss2DsModel.cpp.

References `DefaultPriorType`, `min_sigma`, and `prior_types`.

**9.17.4.27** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

**9.17.4.28** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.17.4.29 ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::get\_num\_pixels ( ) const [inline],  
[inherited]

Definition at line 79 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

Referenced by mappel::ImageFormat2DBase::get\_stats().

**9.17.4.30 StringVecT** mappel::PointEmitterModel::get\_param\_names ( ) const [inline],[inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.17.4.31 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and update\_internal\_1Dsum\_↵  
estimators().

**9.17.4.32 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.17.4.33 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.17.4.34 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( )** [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.17.4.35 const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get\_size ( ) const** [inline],  
[inherited]

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.17.4.36 ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::get\_size ( IdxT *idx* ) const  
[inherited]

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.17.4.37 ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::get\_size\_image\_stack ( const ImageStackT & *stack* ) const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

**9.17.4.38 StatsT** mappel::Gauss2DsModel::get\_stats ( ) const

Definition at line 337 of file Gauss2DsModel.cpp.

References get\_max\_sigma(), get\_max\_sigma\_ratio(), get\_min\_sigma(), mappel::MCMCAdaptor2Ds::get\_stats(), mappel::ImageFormat2DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

**9.17.4.39 const PointEmitterModel::ParamT &** mappel::PointEmitterModel::get\_ubound ( ) const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and set\_max\_sigma\_ratio().

**9.17.4.40 bool** mappel::PointEmitterModel::has\_hyperparam ( const std::string & *name* ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.17.4.41 Gauss2DsModel::Stencil** mappel::Gauss2DsModel::initial\_theta\_estimate ( const ImageT & *im* ) const  
[inline]

Fast, heuristic estimate of initial theta.

Definition at line 224 of file Gauss2DsModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::PointEmitterModel::make\_param().

Referenced by initial\_theta\_estimate().



**9.17.4.42** `Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init ) const` `[inline]`

Definition at line 231 of file Gauss2DsModel.h.

References `mappel::PointEmitterModel::DefaultSeperableInitEstimator`, and `initial_theta_estimate()`.

**9.17.4.43** `Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init, const std::string & estimator ) const`

Definition at line 381 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::Stencil::bg()`, `mappel::methods::estimate_max()`, `mappel::Gauss2DsModel::Stencil::l()`, `mappel::PointEmitterModel::lbound`, `make_stencil()`, `min_sigma`, `mappel::PointEmitterModel::num_params`, `mappel::Gauss2DsModel::Stencil::sigma_ratio()`, `mappel::ImageFormat2DBase::size`, `mappel::estimator::MLEData::theta`, `mappel::PointEmitterModel::theta_in_bounds()`, `mappel::PointEmitterModel::ubound`, `x_model`, and `y_model`.

**9.17.4.44** `CompositeDist mappel::Gauss2DsModel::make_default_prior ( const ImageSizeT & size, double max_sigma_ratio, const std::string & prior_type )` `[static]`

Definition at line 208 of file Gauss2DsModel.cpp.

References `mappel::istarts_with()`, `make_default_prior_beta_position()`, and `make_default_prior_normal_position()`.

Referenced by `mappel::Gauss2DsMAP::Gauss2DsMAP()`, and `mappel::Gauss2DsMLE::Gauss2DsMLE()`.

**9.17.4.45** `CompositeDist mappel::Gauss2DsModel::make_default_prior_beta_position ( const ImageSizeT & size, double max_sigma_ratio )` `[static]`

Definition at line 229 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `make_default_prior()`.

**9.17.4.46** `CompositeDist mappel::Gauss2DsModel::make_default_prior_normal_position ( const ImageSizeT & size, double max_sigma_ratio )` `[static]`

Definition at line 241 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `make_default_prior()`.

**9.17.4.47 ImageFormat2DBase::ImageT** mappel::ImageFormat2DBase::make\_image ( ) const [inline],  
[inherited]

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.17.4.48 ImageFormat2DBase::ImageStackT** mappel::ImageFormat2DBase::make\_image\_stack ( ImageCoordT *n* ) const  
[inline], [inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.17.4.49 Gauss2DsModel::Gauss1DSumModelT** mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator ( IdxT *dim*,  
const ImageSizeT & *size*, const VecT & *min\_sigma*, const VecT & *max\_sigma*, const CompositeDist & *prior* )  
[static], [protected]

Definition at line 69 of file Gauss2DsModel.cpp.

References mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_↵  
position(), min\_sigma, and mappel::ImageFormat2DBase::size.

Referenced by debug\_internal\_sum\_model\_y(), Gauss2DsModel(), and update\_internal\_1Dsum\_estimators().

**9.17.4.50 PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline], [inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(),  
mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DskyModel::initial\_theta\_estimate(), initial\_theta\_↵  
estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::↵  
Gauss2DModel::pixel\_hess\_update(), and pixel\_hess\_update().

**9.17.4.51 template<class FillT > PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( FillT *fill* ) const  
[inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.17.4.52 MatT** mappel::PointEmitterModel::make\_param\_mat ( ) const [inline], [inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.17.4.53 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.17.4.54 `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline]`, `[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.17.4.55 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`  
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.17.4.56 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline]`,  
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.17.4.57 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.17.4.58 `CompositeDist mappel::Gauss2DsModel::make_prior_beta_position ( const ImageSizeT & size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma )` `[static]`

Definition at line 253 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::make\_prior\_component\_intensity(), mappel::PointEmitterModel::make\_prior\_component\_position\_beta(), mappel::PointEmitterModel::make\_prior\_component\_sigma(), and mappel::ImageFormat2DBase::size.

**9.17.4.59** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_I, double kappa = default_intensity_kappa ) [static], [inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `make_prior_normal_position()`.

**9.17.4.60** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos ) [static], [inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `make_prior_beta_position()`.

**9.17.4.61** `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos ) [static], [inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `make_prior_normal_position()`.

**9.17.4.62** `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma ) [static], [inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `make_prior_normal_position()`.

**9.17.4.63** `CompositeDist mappel::Gauss2DsModel::make_prior_normal_position ( const ImageSizeT & size, double sigma_xpos, double sigma_ypos, double mean_I, double kappa_I, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma ) [static]`

Definition at line 268 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

**9.17.4.64** `Gauss2DsModel::Stencil mappel::Gauss2DsModel::make_stencil( const ParamT & theta, bool compute_derivatives = true ) const [inline]`

Make a new Model::Stencil object at theta.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular theta (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not model.theta\_in\_bounds(theta).

If derivatives will not be computed with this stencil set compute\_derivatives=false

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 162 of file Gauss2DsModel.h.

References [mappel::Gauss2DsModel::Stencil::Stencil\(\)](#), and [mappel::PointEmitterModel::theta\\_in\\_bounds\(\)](#).

Referenced by [initial\\_theta\\_estimate\(\)](#).

**9.17.4.65** `Gauss2DsModel & mappel::Gauss2DsModel::operator=( const Gauss2DsModel & o ) [protected]`

Definition at line 44 of file Gauss2DsModel.cpp.

References [get\\_max\\_sigma\(\)](#), [min\\_sigma](#), [mappel::MCMCAdaptor2Ds::operator=\(\)](#), [x\\_model](#), and [y\\_model](#).

Referenced by [debug\\_internal\\_sum\\_model\\_y\(\)](#), [mappel::Gauss2DsMAP::operator=\(\)](#), and [mappel::Gauss2DsMLE::operator=\(\)](#).

**9.17.4.66** `Gauss2DsModel & mappel::Gauss2DsModel::operator=( Gauss2DsModel && o ) [protected]`

Definition at line 56 of file Gauss2DsModel.cpp.

References [min\\_sigma](#), [mappel::MCMCAdaptor2Ds::operator=\(\)](#), [x\\_model](#), and [y\\_model](#).

**9.17.4.67** `void mappel::Gauss2DsModel::pixel_grad( int i, int j, const Stencil & s, ParamT & pgrad ) const [inline]`

Definition at line 180 of file Gauss2DsModel.h.

References [mappel::Gauss2DsModel::Stencil::DX](#), [mappel::Gauss2DsModel::Stencil::DXS](#), [mappel::Gauss2DsModel::Stencil::DY](#), [mappel::Gauss2DsModel::Stencil::DYS](#), [mappel::Gauss2DsModel::Stencil::l\(\)](#), [mappel::Gauss2DsModel::Stencil::X](#), and [mappel::Gauss2DsModel::Stencil::Y](#).

Referenced by [pixel\\_hess\\_update\(\)](#).

**9.17.4.68** `void mappel::Gauss2DsModel::pixel_grad2 ( int i, int j, const Stencil & s, ParamT & pgrad2 ) const [inline]`

Definition at line 192 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::Stencil::DXS`, `mappel::Gauss2DsModel::Stencil::DXS2`, `mappel::Gauss2DsModel::Stencil::DYS`, `mappel::Gauss2DsModel::Stencil::DYS2`, `mappel::Gauss2DsModel::Stencil::l()`, `mappel::Gauss2DsModel::Stencil::sigmaX()`, `mappel::Gauss2DsModel::Stencil::sigmaY()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

**9.17.4.69** `void mappel::Gauss2DsModel::pixel_hess ( int i, int j, const Stencil & s, MatT & hess ) const [inline]`

Definition at line 204 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::Stencil::DX`, `mappel::Gauss2DsModel::Stencil::DXS`, `mappel::Gauss2DsModel::Stencil::DXS2`, `mappel::Gauss2DsModel::Stencil::DXSX`, `mappel::Gauss2DsModel::Stencil::DY`, `mappel::Gauss2DsModel::Stencil::DYS`, `mappel::Gauss2DsModel::Stencil::DYS2`, `mappel::Gauss2DsModel::Stencil::DYSY`, `mappel::Gauss2DsModel::Stencil::l()`, `mappel::Gauss2DsModel::Stencil::sigmaX()`, `mappel::Gauss2DsModel::Stencil::sigmaY()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

**9.17.4.70** `void mappel::Gauss2DsModel::pixel_hess_update ( int i, int j, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess ) const`

pixel derivative inner loop calculations.

Definition at line 354 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::Stencil::DX`, `mappel::Gauss2DsModel::Stencil::DXS`, `mappel::Gauss2DsModel::Stencil::DXS2`, `mappel::Gauss2DsModel::Stencil::DXSX`, `mappel::Gauss2DsModel::Stencil::DY`, `mappel::Gauss2DsModel::Stencil::DYS`, `mappel::Gauss2DsModel::Stencil::DYS2`, `mappel::Gauss2DsModel::Stencil::DYSY`, `mappel::Gauss2DsModel::Stencil::l()`, `mappel::PointEmitterModel::make_param()`, `pixel_grad()`, `mappel::Gauss2DsModel::Stencil::sigmaX()`, `mappel::Gauss2DsModel::Stencil::sigmaY()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

**9.17.4.71** `double mappel::Gauss2DsModel::pixel_model_value ( int i, int j, const Stencil & s ) const [inline]`

Definition at line 173 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::Stencil::bg()`, `mappel::Gauss2DsModel::Stencil::l()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

**9.17.4.72** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const [inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.17.4.73** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.17.4.74** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )` `[inline]`, `[inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.17.4.75** `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 56 of file `MCMCAdaptor2Ds.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor2Ds::eta_↵_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_↵_phases`, and `mappel::rng_manager`.

**9.17.4.76** `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const` `[inherited]`

Definition at line 75 of file `MCMCAdaptor2Ds.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor2Ds::eta_↵_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_↵_phases`, and `mappel::rng_manager`.

**9.17.4.77** `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const` `[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.17.4.78** `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline]`, `[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.17.4.79 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 ) [inherited]`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.17.4.80 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ ) [inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.17.4.81 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc ) [inline], [inherited]`

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.17.4.82 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.17.4.83 `void mappel::Gauss2DsModel::set_hyperparams ( const VecT & hyperparams )`

Definition at line 119 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::set_hyperparams()`, and `update_internal_1Dsum_estimators()`.

9.17.4.84 `template<class ImT > void mappel::ImageFormat2DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]`

Definition at line 113 of file ImageFormat2DBase.h.



9.17.4.85 void mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling ( double *eta\_l* = -1 ) [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_mean\_l, mappel::MCMCAdaptor1D::eta\_l, mappel::PointEmitterModel::get\_hyperparam\_value(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

9.17.4.86 void mappel::PointEmitterModel::set\_lbound ( const ParamT & *lbound* ) [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.17.4.87 void mappel::Gauss2DsModel::set\_max\_sigma ( const VecT & *new\_sigma* )

Set the max\_sigma\_ratio based on the new max\_sigma's ratio with the current min\_sigma.

Definition at line 155 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::check\_psf\_sigma(), compute\_max\_sigma\_ratio(), get\_min\_sigma(), and set\_max\_sigma\_ratio().

9.17.4.88 void mappel::Gauss2DsModel::set\_max\_sigma\_ratio ( double *max\_sigma\_ratio* )

Definition at line 176 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, get\_max\_sigma(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::PointEmitterModel::set\_ubound(), x\_model, and y\_model.

Referenced by set\_max\_sigma().

9.17.4.89 void mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases ( IdxT *num\_phases* ) [protected],[inherited]

Definition at line 59 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

Referenced by mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), and mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds().

9.17.4.90 void mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale ( double *scale* ) [inherited]

Definition at line 39 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::global\_max\_mcmc\_sigma\_scale, and mappel::MCMCAdaptorBase::sigma\_scale.

9.17.4.91 `void mappel::Gauss2DsModel::set_min_sigma ( const VecT & new_sigma )`

Set the minimum sigma, keeping the max\_sigma\_ratio the same.

Definition at line 137 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, `get_max_sigma()`, `min_sigma`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `x_model`, and `y_model`.

9.17.4.92 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` `[inline]`, `[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.17.4.93 `void mappel::Gauss2DsModel::set_prior ( CompositeDist && prior_ )`

Definition at line 107 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `update_internal_1Dsum_estimators()`.

9.17.4.94 `void mappel::Gauss2DsModel::set_prior ( const CompositeDist & prior_ )`

Definition at line 113 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::set_prior()`, and `update_internal_1Dsum_estimators()`.

9.17.4.95 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` `[static]`, `[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.17.4.96 `void mappel::Gauss2DsModel::set_size ( const ImageSizeT & size_ )`

Definition at line 125 of file Gauss2DsModel.cpp.

References `mappel::ImageFormat2DBase::set_size()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::ImageFormat2DBase::size`, `x_model`, and `y_model`.

9.17.4.97 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `set_max_sigma_ratio()`.

9.17.4.98 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const` `[inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxyModel::make_stencil()`, `make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

9.17.4.99 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

9.17.4.100 `void mappel::Gauss2DsModel::update_internal_1Dsum_estimators ( )` `[protected]`

Definition at line 100 of file Gauss2DsModel.cpp.

References `get_max_sigma()`, `mappel::PointEmitterModel::get_prior()`, `make_internal_1Dsum_estimator()`, `min_sigma`, `mappel::ImageFormat2DBase::size`, `x_model`, and `y_model`.

Referenced by `debug_internal_sum_model_y()`, `set_hyperparams()`, and `set_prior()`.

## 9.17.5 Member Data Documentation

9.17.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static]`, `[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

9.17.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static]`, `[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.17.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static]`, `[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

**9.17.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.17.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.17.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.17.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.17.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.17.5.9** `const std::string mappel::Gauss2DsModel::DefaultPriorType = "Normal"` `[static]`

Definition at line 59 of file Gauss2DsModel.h.

Referenced by `get_min_sigma()`.

**9.17.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],  
[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), and initial\_theta\_estimate().

**9.17.5.11** `double mappel::MCMCAdaptor1D::eta_bg =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to theta\_bg in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by mappel::MCMCAdaptor1D::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1D::operator=(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.17.5.12** `double mappel::MCMCAdaptor1D::eta_I =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to theta\_I in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by mappel::MCMCAdaptor1D::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1D::operator=(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.17.5.13** `double mappel::MCMCAdaptor2Ds::eta_sigma =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to theta\_bg in the random walk MCMC sampling

Definition at line 27 of file MCMCAdaptor2Ds.h.

Referenced by mappel::MCMCAdaptor2Ds::get\_stats(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor2Ds::operator=(), and mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate().

**9.17.5.14** `double mappel::MCMCAdaptor1D::eta_x =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to theta\_x in the random walk MCMC sampling

Definition at line 31 of file MCMCAdaptor1D.h.

Referenced by mappel::MCMCAdaptor1D::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1D::operator=(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), and mappel::MCMCAdaptor1D::sample\_mcmc\_candidate().

**9.17.5.15** `double mappel::MCMCAdaptor2D::eta_y = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor2D.h`.

Referenced by `mappel::MCMCAdaptor2D::get_stats()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2D::operator=()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`.

**9.17.5.16** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.17.5.17** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.17.5.18** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.17.5.19** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file `ImageFormat2DBase.h`.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.17.5.20** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.17.5.21** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size =3` `[static]`,  
`[inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::check\_size().

**9.17.5.22** `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

**9.17.5.23** `VecT mappel::Gauss2DsModel::min_sigma` `[protected]`

Gaussian PSF in pixels

Definition at line 118 of file Gauss2DsModel.h.

Referenced by compute\_max\_sigma\_ratio(), debug\_internal\_sum\_model\_y(), mappel::Gauss2DsMAP::Gauss2DsMAP(), mappel::Gauss2DsMLE::Gauss2DsMLE(), Gauss2DsModel(), get\_min\_sigma(), initial\_theta\_estimate(), make\_internal\_1Dsum\_estimator(), operator=(), set\_min\_sigma(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), and update\_internal\_1Dsum\_estimators().

**9.17.5.24** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim =2` `[static]`,  
`[inherited]`

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::get\_stats().

**9.17.5.25** `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.17.5.26 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.17.5.27 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

#### 9.17.5.28 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.17.5.29 `const StringVecT mappel::Gauss2DsModel::prior_types` `[static]`

**Initial value:**

```
= { "Beta",  
    "Normal"  
}
```

Definition at line 58 of file `Gauss2DsModel.h`.

Referenced by `get_min_sigma()`.



### 9.17.5.30 double mappel::MCMCAdaptorBase::sigma\_scale [protected], [inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

### 9.17.5.31 ImageSizeT mappel::ImageFormat2DBase::size [protected], [inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_heuristic\_compute\_estimate(), mappel::ImageFormat2DBase::check\_image\_shape(), mappel::Gauss2DModel::Stencil::compute\_derivatives(), mappel::Gauss2DsModel::Stencil::compute\_derivatives(), mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), debug\_internal\_sum\_model\_y(), mappel::methods::expected\_information(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsMAP::Gauss2DsMAP(), mappel::Gauss2DsMLE::Gauss2DsMLE(), Gauss2DsModel(), mappel::ImageFormat2DBase::get\_num\_pixels(), mappel::ImageFormat2DBase::get\_size(), mappel::ImageFormat2DBase::get\_stats(), mappel::methods::likelihood::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood::debug::grad\_components(), mappel::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian\_components(), mappel::Gauss2DModel::initial\_theta\_estimate(), initial\_theta\_estimate(), mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh\_components(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), make\_default\_prior\_normal\_position(), mappel::ImageFormat2DBase::make\_image(), mappel::ImageFormat2DBase::make\_image\_stack(), mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator(), make\_internal\_1Dsum\_estimator(), mappel::Gauss2DModel::make\_prior\_beta\_position(), make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), make\_prior\_normal\_position(), mappel::methods::model\_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods::likelihood::rllh(), mappel::methods::likelihood::debug::rllh\_components(), mappel::ImageFormat2DBase::set\_size(), mappel::Gauss2DModel::set\_size(), set\_size(), mappel::methods::simulate\_image(), mappel::methods::simulate\_image\_from\_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2DsModel::Stencil::Stencil(), mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and update\_internal\_1Dsum\_estimators().

### 9.17.5.32 ParamT mappel::PointEmitterModel::ubound [protected], [inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

### 9.17.5.33 Gauss1DSumModelT mappel::Gauss2DsModel::x\_model [protected]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 119 of file Gauss2DsModel.h.

Referenced by debug\_internal\_sum\_model\_x(), initial\_theta\_estimate(), operator=(), set\_max\_sigma\_ratio(), set\_min\_sigma\_ratio(), set\_size(), and update\_internal\_1Dsum\_estimators().

### 9.17.5.34 Gauss1DSumModelT mappel::Gauss2DsModel::y\_model [protected]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 120 of file Gauss2DsModel.h.

Referenced by debug\_internal\_sum\_model\_y(), Gauss2DsModel(), initial\_theta\_estimate(), operator=(), set\_max\_sigma\_ratio(), set\_min\_sigma\_ratio(), set\_size(), and update\_internal\_1Dsum\_estimators().

The documentation for this class was generated from the following files:

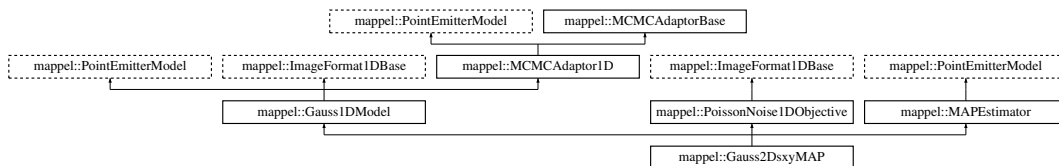
- [Gauss2DsModel.h](#)
- [Gauss2DsModel.cpp](#)

## 9.18 mappel::Gauss2DsxyMAP Class Reference

A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DsxyMAP.h>
```

Inheritance diagram for mappel::Gauss2DsxyMAP:



### Public Types

- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = CoordT
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Col< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Col< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Mat< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >
- using [ModelDataT](#) = [ImageT](#)
- using [ModelDataStackT](#) = [ImageStackT](#)

## Public Member Functions

- [Gauss2DsxMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma)
- [Gauss2DsxMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma, CompositeDist &&prior)
- double [get\\_psf\\_sigma](#) () const
- double [get\\_psf\\_sigma](#) ([IdxT](#) idx) const
- void [set\\_psf\\_sigma](#) (double new\_psf\_sigma)
- void [set\\_psf\\_sigma](#) (const [VecT](#) &new\_psf\_sigma)
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &theta, bool compute\_derivatives=true) const  
*Make a new Model::Stencil object at theta.*
- double [pixel\\_model\\_value](#) ([IdxT](#) i, const [Stencil](#) &s) const
- void [pixel\\_grad](#) ([IdxT](#) i, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel\\_grad2](#) ([IdxT](#) i, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel\\_hess](#) ([IdxT](#) i, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel\\_hess\\_update](#) ([IdxT](#) i, const [Stencil](#) &s, double dm\_ratio\_m1, double dmm\_ratio, [ParamT](#) &grad, [MatT](#) &hess) const  
*pixel derivative inner loop calculations.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im) const  
*Fast, heuristic estimate of initial theta.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init) const
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT](#) [make\\_param](#) () const
- template<class FillT >  
[ParamT](#) [make\\_param](#) (FillT fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
[ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT](#) [make\\_param\\_mat](#) () const
- template<class FillT >  
[MatT](#) [make\\_param\\_mat](#) (FillT fill) const
- [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
[CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get\\_prior](#) ()
- const CompositeDist & [get\\_prior](#) () const
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- [IdxT](#) [get\\_num\\_hyperparams](#) () const
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- [VecT](#) [get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)

- [StringVecT get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT sample\\_prior](#) (RngT &rng) const
- [ParamT sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) &[get\\_lbound](#) () const
- const [ParamT](#) &[get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT make\\_image](#) () const
- [ImageStackT make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- template<class ImT >  
  void [set\\_image\\_in\\_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const ImT &im) const
- [ImageSizeT get\\_size](#) () const
- [ImageCoordT get\\_size](#) ([IdxT](#) idx) const
- [ImageCoordT get\\_num\\_pixels](#) () const
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- void [set\\_size](#) (const arma::Col< [ImageCoordT](#) > &sz)
- void [check\\_image\\_shape](#) (const [ImageT](#) &im) const  
  *Check the shape of a single images is correct for model size.*
- void [check\\_image\\_shape](#) (const [ImageStackT](#) &ims) const  
  *Check the shape of a stack of images is correct for model size.*
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, const [IdxVecT](#) &fixed\_parameters\_↵  
  mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)
- void [set\\_background\\_mcmc\\_sampling](#) (double eta\_bg=-1)
- void [set\\_mcmc\\_sigma\\_scale](#) (double scale)
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- [IdxT get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Member Functions

- static CompositeDist [make\\_default\\_prior](#) ([IdxT](#) size, const std::string &prior\_type)
- static CompositeDist [make\\_default\\_prior\\_beta\\_position](#) ([IdxT](#) size)
- static CompositeDist [make\\_default\\_prior\\_normal\\_position](#) ([IdxT](#) size)
- static CompositeDist [make\\_prior\\_beta\\_position](#) ([IdxT](#) size, double beta\_xpos, double mean\_l, double kappa\_l,  
  double mean\_bg, double kappa\_bg)

- static CompositeDist [make\\_prior\\_normal\\_position](#) (IdxT size, double sigma\_xpos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg)
- static prior\_hessian::TruncatedNormalDist [make\\_prior\\_component\\_position\\_normal](#) (IdxT size, double pos\_↵ sigma=[default\\_sigma\\_pos](#))
- static prior\_hessian::ScaledSymmetricBetaDist [make\\_prior\\_component\\_position\\_beta](#) (IdxT size, double pos\_↵ beta=[default\\_beta\\_pos](#))
- static prior\_hessian::TruncatedGammaDist [make\\_prior\\_component\\_intensity](#) (double mean=[default\\_mean\\_↵ l](#), double kappa=[default\\_intensity\\_kappa](#))
- static prior\_hessian::TruncatedParetoDist [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵ sigma, double alpha=[default\\_alpha\\_sigma](#))
- static void [set\\_rng\\_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)

*Check the size argument for the model.*

#### Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior\\_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) = 1
- static const [ImageCoordT](#) [global\\_min\\_size](#) = 3
- static const [ImageCoordT](#) [global\\_max\\_size](#) = 512
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5
- static const std::vector< std::string > [estimator\\_names](#)

#### Protected Member Functions

- void [set\\_mcmc\\_num\\_phases](#) (IdxT num\_phases)

### Protected Attributes

- double [psf\\_sigma](#)
- CompositeDist [prior](#)
- [IdxT num\\_params](#)
- [IdxT num\\_hyperparams](#)
- [ParamT lbound](#)
- [ParamT ubound](#)
- [ImageSizeT size](#)
- double [eta\\_x](#) =0
- double [eta\\_l](#) =0
- double [eta\\_bg](#) =0
- [IdxT num\\_phases](#)
- double [sigma\\_scale](#)

### 9.18.1 Detailed Description

A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.

Model: [Gauss1DModel](#) a 1D gaussian PSF with fixed psf\_sigma Objective Statistical Noise Model: PoissonNoise1D↔  
MAPObjective an MLE objective for Poisson noise ImageFormat: [ImageFormat1DBase](#) - Data format

Definition at line 25 of file Gauss2DsxyMAP.h.

### 9.18.2 Member Typedef Documentation

**9.18.2.1** `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` `[inherited]`

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

**9.18.2.2** `using mappel::ImageFormat1DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

**9.18.2.3** `template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT>`  
`[inherited]`

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

9.18.2.4 `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT` [inherited]

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

9.18.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`  
[inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

9.18.2.6 `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT>`  
[inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

9.18.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
[inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

9.18.2.8 `template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>`  
[inherited]

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

9.18.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
[inherited]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

9.18.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

9.18.2.11 **using mappel::PoissonNoise1DObjective::ModelDataStackT = ImageStackT** [inherited]

Objective function data stack type: 1D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 26 of file PoissonNoise1DObjective.h.

9.18.2.12 **using mappel::PoissonNoise1DObjective::ModelDataT = ImageT** [inherited]

Objective function data type: 1D double precision image, gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise1DObjective.h.

9.18.2.13 **using mappel::PointEmitterModel::ParamT = arma::vec** [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.18.2.14 **using mappel::PointEmitterModel::ParamVecT = arma::mat** [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

9.18.2.15 **using mappel::Gauss1DModel::StencilVecT = std::vector<Stencil>** [inherited]

Definition at line 49 of file Gauss1DModel.h.

### 9.18.3 Constructor & Destructor Documentation

9.18.3.1 **mappel::Gauss2DskyMAP::Gauss2DskyMAP ( const ImageSizeT & size, const VecT & min\_sigma, const VecT & max\_sigma )**

9.18.3.2 **mappel::Gauss2DskyMAP::Gauss2DskyMAP ( const ImageSizeT & size, const VecT & min\_sigma, const VecT & max\_sigma, CompositeDist && prior )**

### 9.18.4 Member Function Documentation

9.18.4.1 **void mappel::PointEmitterModel::bound\_theta ( ParamT & theta, double epsilon = bounds\_epsilon ) const**  
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.



**9.18.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const** [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

**9.18.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & *theta*, double *epsilon* = bounds\_epsilon ) const** [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

**9.18.4.4 void ImageFormat1DBase::check\_image\_shape ( const ImageT & *im* ) const** [inherited]

Check the shape of a single images is correct for model size.

Definition at line 59 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

**9.18.4.5 void ImageFormat1DBase::check\_image\_shape ( const ImageStackT & *ims* ) const** [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

**9.18.4.6 void mappel::PointEmitterModel::check\_param\_shape ( const ParamT & *theta* ) const** [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

**9.18.4.7 void mappel::PointEmitterModel::check\_param\_shape ( const ParamVecT & *theta* ) const** [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

**9.18.4.8** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` `[inherited]`

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

**9.18.4.9** `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

**9.18.4.10** `void ImageFormat1DBase::check_size ( const ImageSizeT & size )` `[static]`, `[inherited]`

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::global_max_size`, and `mappel::ImageFormat1DBase::global_min_size`.

Referenced by `mappel::ImageFormat1DBase::ImageFormat1DBase()`, and `mappel::ImageFormat1DBase::set_size()`.

**9.18.4.11** `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline]`, `[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.18.4.12** `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` `[inline]`, `[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.18.4.13** `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline]`, `[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.18.4.14 **PointEmitterModel::ParamT** mappel::PointEmitterModel::get\_hyperparams ( ) const [inline],  
[inherited]

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.18.4.15 **ImageFormat1DBase::ImageT** ImageFormat1DBase::get\_image\_from\_stack ( const ImageStackT & stack,  
ImageCoordT n ) const [inline], [inherited]

Definition at line 108 of file ImageFormat1DBase.h.

9.18.4.16 **const PointEmitterModel::ParamT &** mappel::PointEmitterModel::get\_lbound ( ) const [inline],  
[inherited]

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

9.18.4.17 **IdxT** mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases ( ) const [inherited]

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

9.18.4.18 **double** mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale ( ) const [inherited]

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

9.18.4.19 **IdxT** mappel::PointEmitterModel::get\_num\_hyperparams ( ) const [inline], [inherited]

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

9.18.4.20 **IdxT** mappel::PointEmitterModel::get\_num\_params ( ) const [inline], [inherited]

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.18.4.21 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get\_num\_pixels ( ) const** [inline], [inherited]

Definition at line 82 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.18.4.22 StringVecT mappel::PointEmitterModel::get\_param\_names ( ) const** [inline], [inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.18.4.23 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline], [inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.18.4.24 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline], [inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.18.4.25 double mappel::Gauss1DModel::get\_psf\_sigma ( ) const** [inline], [inherited]

Definition at line 127 of file Gauss1DModel.h.

References mappel::Gauss1DModel::psf\_sigma.

Referenced by mappel::Gauss1DModel::get\_stats().

**9.18.4.26 double mappel::Gauss1DModel::get\_psf\_sigma ( IdxT idx ) const** [inherited]

Definition at line 131 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::psf\_sigma.

**9.18.4.27 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static], [inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.18.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( )** [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.18.4.29 ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get\_size ( ) const** [inline],[inherited]

Definition at line 71 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

**9.18.4.30 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get\_size ( IdxT idx ) const** [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

**9.18.4.31 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get\_size\_image\_stack ( const ImageStackT & stack ) const** [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

**9.18.4.32 StatsT mappel::Gauss1DModel::get\_stats ( ) const** [inherited]

Definition at line 178 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::get\_psf\_sigma(), mappel::MCMCAdaptor1D::get\_stats(), mappel::ImageFormat1DBase::get\_stats(), and mappel::PointEmitterModel::get\_stats().

**9.18.4.33 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const** [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

**9.18.4.34 bool mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const** [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.18.4.35 Gauss1DModel::Stencil mappel::Gauss1DModel::initial\_theta\_estimate ( const ImageT & *im* ) const** [inline], [inherited]

Fast, heuristic estimate of initial theta.

Definition at line 169 of file Gauss1DModel.h.

References mappel::PointEmitterModel::make\_param(), and mappel::Gauss1DModel::Stencil::theta.

**9.18.4.36 Gauss1DModel::Stencil mappel::Gauss1DModel::initial\_theta\_estimate ( const ImageT & *im*, const ParamT & *theta\_init* ) const** [inherited]

Definition at line 207 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::Stencil::bg(), mappel::Gauss1DModel::Stencil::l(), mappel::Gauss1DModel::make\_stencil(), mappel::PointEmitterModel::num\_params, and mappel::ImageFormat1DBase::size.

**9.18.4.37 CompositeDist mappel::Gauss1DModel::make\_default\_prior ( IdxT *size*, const std::string & *prior\_type* )** [static], [inherited]

Definition at line 59 of file Gauss1DModel.cpp.

References mappel::istarts\_with(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), and mappel::Gauss1DModel::make\_default\_prior\_normal\_position().

**9.18.4.38 CompositeDist mappel::Gauss1DModel::make\_default\_prior\_beta\_position ( IdxT *size* )** [static], [inherited]

Definition at line 80 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_beta().

Referenced by mappel::Gauss1DModel::make\_default\_prior().

**9.18.4.39 CompositeDist mappel::Gauss1DModel::make\_default\_prior\_normal\_position ( IdxT *size* )** [static], [inherited]

Definition at line 90 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::PointEmitterModel::make\_prior\_component\_intensity(), and mappel::PointEmitterModel::make\_prior\_component\_position\_normal().

Referenced by mappel::Gauss1DModel::make\_default\_prior().

**9.18.4.40 ImageFormat1DBase::ImageT ImageFormat1DBase::make\_image ( ) const** [inline], [inherited]

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.18.4.41 ImageFormat1DBase::ImageStackT ImageFormat1DBase::make\_image\_stack ( ImageCoordT *n* ) const**  
 [inline],[inherited]

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

**9.18.4.42 PointEmitterModel::ParamT mappel::PointEmitterModel::make\_param ( ) const** [inline],[inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

**9.18.4.43 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make\_param ( FillT *fill* ) const**  
 [inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.18.4.44 MatT mappel::PointEmitterModel::make\_param\_mat ( ) const** [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.18.4.45 template<class FillT > MatT mappel::PointEmitterModel::make\_param\_mat ( FillT *fill* ) const** [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.18.4.46 CubeT mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n* ) const** [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.18.4.47 template<class FillT > CubeT mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n*, FillT *fill* ) const**  
 [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.18.4.48** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const [inline], [inherited]`

Definition at line 168 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.18.4.49** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const [inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.18.4.50** `CompositeDist mappel::Gauss1DModel::make_prior_beta_position ( IdxT size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg ) [static], [inherited]`

Definition at line 101 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_beta()`.

Referenced by `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`.

**9.18.4.51** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_l, double kappa = default_intensity_kappa ) [static], [inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.18.4.52** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos ) [static], [inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.



9.18.4.53 **prior\_hessian::TruncatedNormalDist** `mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos )` [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.18.4.54 **prior\_hessian::TruncatedParetoDist** `mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma )` [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.18.4.55 **CompositeDist** `mappel::Gauss1DModel::make_prior_normal_position ( IdxT size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg )` [static],[inherited]

Definition at line 114 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_normal()`.

Referenced by `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`.

9.18.4.56 **Gauss1DModel::Stencil** `mappel::Gauss1DModel::make_stencil ( const ParamT & theta, bool compute_derivatives = true ) const` [inline],[inherited]

Make a new Model::Stencil object at theta.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular theta (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

## Returns

A new [Stencil](#) object ready to compute with

Definition at line 116 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`.

```
9.18.4.57 void mappel::Gauss1DModel::pixel_grad ( IdxT i, const Stencil & s, ParamT & pgrad ) const [inline],
[inherited]
```

Definition at line 141 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::DX`, `mappel::Gauss1DModel::Stencil::l()`, and `mappel::Gauss1DModel::Stencil::X`.

Referenced by `mappel::Gauss1DModel::pixel_hess_update()`.

```
9.18.4.58 void mappel::Gauss1DModel::pixel_grad2 ( IdxT i, const Stencil & s, ParamT & pgrad2 ) const [inline],
[inherited]
```

Definition at line 150 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::DXS`, `mappel::Gauss1DModel::Stencil::l()`, and `mappel::Gauss1DModel::psf_sigma`.

```
9.18.4.59 void mappel::Gauss1DModel::pixel_hess ( IdxT i, const Stencil & s, MatT & hess ) const [inline],
[inherited]
```

Definition at line 159 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::DX`, `mappel::Gauss1DModel::Stencil::DXS`, `mappel::Gauss1DModel::Stencil::l()`, and `mappel::Gauss1DModel::psf_sigma`.

```
9.18.4.60 void mappel::Gauss1DModel::pixel_hess_update ( IdxT i, const Stencil & s, double dm_ratio_m1, double dmm_ratio,
ParamT & grad, MatT & hess ) const [inherited]
```

pixel derivative inner loop calculations.

Definition at line 191 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::Stencil::DXS`, `mappel::Gauss1DModel::Stencil::l()`, `mappel::PointEmitterModel::make_param()`, `mappel::Gauss1DModel::pixel_grad()`, and `mappel::Gauss1DModel::psf_sigma`.

```
9.18.4.61 double mappel::Gauss1DModel::pixel_model_value ( IdxT i, const Stencil & s ) const [inline],
[inherited]
```

Definition at line 135 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::Stencil::bg()`, `mappel::Gauss1DModel::Stencil::l()`, and `mappel::Gauss1DModel::Stencil::X`.

**9.18.4.62** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.18.4.63** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const`  
`[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.18.4.64** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.18.4.65** `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const`  
`[inherited]`

Definition at line 108 of file MCMCAdaptor1D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.18.4.66** `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const`  
`[inherited]`

Definition at line 122 of file MCMCAdaptor1D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.18.4.67** `template<class RngT> PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.18.4.68** `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline],[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

**9.18.4.69** `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` `[inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

**9.18.4.70** `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )` `[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.18.4.71** `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline],[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.18.4.72** `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline],[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.18.4.73** `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams )` `[inline],[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

9.18.4.74 `template<class ImT > void ImageFormat1DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const` [inherited]

Definition at line 115 of file ImageFormat1DBase.h.

9.18.4.75 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 )` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.18.4.76 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.18.4.77 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` [protected],[inherited]

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.18.4.78 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` [inherited]

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.18.4.79 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.18.4.80 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )` [inherited]

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

**9.18.4.81** `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ )` `[inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.18.4.82** `void mappel::Gauss1DModel::set_psf_sigma ( double new_psf_sigma )` `[inherited]`

Definition at line 125 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`, and `mappel::Gauss1DModel::psf_sigma`.

Referenced by `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

**9.18.4.83** `void mappel::Gauss1DModel::set_psf_sigma ( const VecT & new_psf_sigma )` `[inline],[inherited]`

Definition at line 131 of file Gauss1DModel.h.

References `mappel::Gauss1DModel::set_psf_sigma()`.

**9.18.4.84** `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` `[static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

**9.18.4.85** `void ImageFormat1DBase::set_size ( const ImageSizeT & size_ )` `[inherited]`

Definition at line 30 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::check_size()`, and `mappel::ImageFormat1DBase::size`.

Referenced by `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss2DModel::set_size()`, and `mappel::Gauss2DsModel::set_size()`.

**9.18.4.86** `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz )` `[inline],[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

**9.18.4.87** `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

9.18.4.88 **bool** mappel::PointEmitterModel::theta\_in\_bounds ( const ParamT & *theta* ) const [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::make\_stencil(), mappel::Gauss1DsModel::make\_stencil(), mappel::Gauss2DModel::make\_stencil(), mappel::Gauss2DsxyModel::make\_stencil(), mappel::Gauss2DsModel::make\_stencil(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.18.4.89 **BoolVecT** mappel::PointEmitterModel::theta\_stack\_in\_bounds ( const ParamVecT & *theta* ) const [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::theta\_in\_bounds().

## 9.18.5 Member Data Documentation

9.18.5.1 **const double** mappel::PointEmitterModel::bounds\_epsilon = 1.0E-6 [static], [inherited]

Distance from the boundary to constrain in bound\_theta and bounded\_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), and mappel::PointEmitterModel::set\_ubound().

9.18.5.2 **const double** mappel::PointEmitterModel::default\_alpha\_sigma = 2 [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.18.5.3 **const double** mappel::PointEmitterModel::default\_beta\_pos = 3 [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.18.5.4 **const double** mappel::PointEmitterModel::default\_intensity\_kappa = 2 [static], [inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.18.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.18.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.18.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.18.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.18.5.9** `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 53 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::operator=()`.

**9.18.5.10** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.



9.18.5.11 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names` `[static], [inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

9.18.5.12 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.18.5.13 `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.18.5.14 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

9.18.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.18.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.18.5.17** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.18.5.18** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.18.5.19** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.18.5.20** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

**9.18.5.21** `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.18.5.22** `const std::string mappel::Gauss2DsxyMAP::name` `[static]`

Definition at line 30 of file `Gauss2DsxyMAP.h`.

9.18.5.23 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static]`, `[inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::get\_stats().

9.18.5.24 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

9.18.5.25 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

9.18.5.26 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().

9.18.5.27 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

**9.18.5.28** `const StringVect mappel::Gauss1DModel::prior_types` `[static],[inherited]`

**Initial value:**

```
= { "Beta",
                                     "Normal"
}
```

Definition at line 52 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::operator=()`.

**9.18.5.29** `double mappel::Gauss1DModel::psf_sigma` `[protected],[inherited]`

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 90 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::get_psf_sigma()`, `mappel::Gauss1DModel::operator=()`, `mappel::Gauss1DModel::pixel_grad2()`, `mappel::Gauss1DModel::pixel_hess()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

**9.18.5.30** `double mappel::MCMCAdaptorBase::sigma_scale` `[protected],[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.18.5.31** `ImageSizeT mappel::ImageFormat1DBase::size` `[protected],[inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

## 9.18.5.32 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

The documentation for this class was generated from the following file:

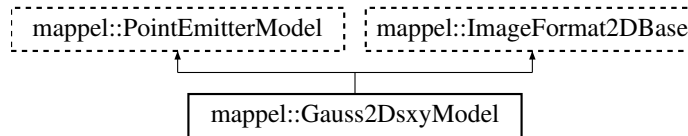
- [Gauss2DsxxyMAP.h](#)

## 9.19 mappel::Gauss2DsxxyModel Class Reference

A base class for 2D Gaussian PSF with axis-aligned gaussian with free parameters for both sigma\_x and sigma\_y. Gaussian sigma parameters sigma\_x and sigma\_y are measured in units of pixels. The model has 6 parameters, [x,y,l,bg,sigma\_x,sigma\_y].

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DsxxyModel.h>
```

Inheritance diagram for mappel::Gauss2DsxxyModel:



## Classes

- class [Stencil](#)  
*Stencil for 2D free-sigma (astigmatic) models.*

## Public Types

- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >

## Public Member Functions

- [Gauss2DsxxyModel](#) (const [ImageSizeT](#) &size, const [VecT](#) &min\_sigma, const [VecT](#) &max\_sigma)
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- [VecT](#) [get\\_min\\_sigma](#) () const
- double [get\\_min\\_sigma](#) ([IdxT](#) dim) const
- [VecT](#) [get\\_max\\_sigma](#) () const
- double [get\\_max\\_sigma](#) ([IdxT](#) dim) const
- double [get\\_max\\_sigma\\_ratio](#) () const
- void [set\\_min\\_sigma](#) (const [VecT](#) &min\_sigma)
- void [set\\_max\\_sigma](#) (const [VecT](#) &max\_sigma)
- void [set\\_max\\_sigma\\_ratio](#) (double max\_sigma\_ratio)
- [StatsT](#) [get\\_stats](#) () const
- [Stencil](#) [make\\_stencil](#) (const [ParamT](#) &theta, bool compute\_derivatives=true) const  
*Make a new Model::Stencil object at theta.*
- double [pixel\\_model\\_value](#) (int i, int j, const [Stencil](#) &s) const
- void [pixel\\_grad](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel\\_grad2](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel\\_hess](#) (int i, int j, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel\\_hess\\_update](#) (int i, int j, const [Stencil](#) &s, double dm\_ratio\_m1, double dmm\_ratio, [ParamT](#) &grad, [MatT](#) &hess) const
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im)  
*Fast, heuristic estimate of initial theta.*
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init)
- [Stencil](#) [initial\\_theta\\_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta\_init, const std::string &estimator)
- void [sample\\_mcmc\\_candidate](#) (int sample\_index, [ParamT](#) &candidate\_theta, double scale=1.0)
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT](#) [make\\_param](#) () const
- template<class FillT >  
[ParamT](#) [make\\_param](#) (FillT fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
[ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT](#) [make\\_param\\_mat](#) () const
- template<class FillT >  
[MatT](#) [make\\_param\\_mat](#) (FillT fill) const
- [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
[CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get\\_prior](#) ()
- const CompositeDist & [get\\_prior](#) () const
- void [set\\_prior](#) (const CompositeDist &prior\_)
- [IdxT](#) [get\\_num\\_hyperparams](#) () const
- [VecT](#) [get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const

- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT](#) [get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT](#) [get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT](#) [sample\\_prior](#) (RngT &rng) const
- [ParamT](#) [sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get\\_lbound](#) () const
- const [ParamT](#) & [get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT](#) [bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT](#) [reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT](#) [theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT](#) [bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT](#) [reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- [ImageT](#) [make\\_image](#) () const
- [ImageStackT](#) [make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT](#) [get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT](#) [get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- template<class ImT >  
  void [set\\_image\\_in\\_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const ImT &im) const
- const [ImageSizeT](#) & [get\\_size](#) () const
- [ImageCoordT](#) [get\\_size](#) ([IdxT](#) idx) const
- [ImageCoordT](#) [get\\_num\\_pixels](#) () const
- void [check\\_image\\_shape](#) (const [ImageT](#) &im) const  
  *Check the shape of a single images is correct for model size.*
- void [check\\_image\\_shape](#) (const [ImageStackT](#) &ims) const  
  *Check the shape of a stack of images is correct for model size.*

#### Static Public Member Functions

- static CompositeDist [make\\_default\\_prior](#) (const [ImageSizeT](#) &size, double max\_sigma\_ratio)
- static CompositeDist [make\\_prior\\_beta\\_position](#) (const [ImageSizeT](#) &size, double beta\_xpos, double beta\_ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double max\_sigma\_ratio, double alpha\_←sigma)
- static CompositeDist [make\\_prior\\_normal\\_position](#) (const [ImageSizeT](#) &size, double sigma\_xpos, double sigma\_←ypos, double mean\_l, double kappa\_l, double mean\_bg, double kappa\_bg, double max\_sigma\_ratio, double alpha\_sigma)
- static prior\_hessian::TruncatedNormalDist [make\\_prior\\_component\\_position\\_normal](#) ([IdxT](#) size, double pos\_←sigma=default\_sigma\_pos)
- static prior\_hessian::ScaledSymmetricBetaDist [make\\_prior\\_component\\_position\\_beta](#) ([IdxT](#) size, double pos\_←beta=default\_beta\_pos)

- static prior\_hessian::TruncatedGammaDist [make\\_prior\\_component\\_intensity](#) (double mean=[default\\_mean\\_↵](#), double kappa=[default\\_intensity\\_kappa](#))
- static prior\_hessian::TruncatedParetoDist [make\\_prior\\_component\\_sigma](#) (double [min\\_sigma](#), double max\_[↵](#)sigma, double alpha=[default\\_alpha\\_sigma](#))
- static void [set\\_rng\\_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()
- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)

*Check the size argument for the model.*

#### Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const [ImageCoordT](#) [num\\_dim](#) =2
- static const [ImageCoordT](#) [global\\_min\\_size](#) =3
- static const [ImageCoordT](#) [global\\_max\\_size](#) =512

#### Protected Member Functions

- void [update\\_internal\\_1D\\_estimators](#) ()

#### Static Protected Member Functions

- static double [compute\\_max\\_sigma\\_ratio](#) (const [VecT](#) &[min\\_sigma](#), const [VecT](#) &[max\\_sigma](#))

#### Protected Attributes

- double [mcmc\\_candidate\\_eta\\_y](#)
- double [mcmc\\_candidate\\_eta\\_sigma](#)
- [VecT](#) [min\\_sigma](#)
- [Gauss1DsMAP](#) [x\\_model](#)
- [Gauss1DsMAP](#) [y\\_model](#)
- CompositeDist [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)



### 9.19.1 Detailed Description

A base class for 2D Gaussian PSF with axis-aligned gaussian with free parameters for both `sigma_x` and `sigma_y`. Gaussian sigma parameters `sigma_x` and `sigma_y` are measured in units of pixels. The model has 6 parameters, `[x,y,l,bg,sigma_x,sigma_y]`.

Importantly `sigma_x` and `sigma_y` must be in the range given by parameters `min_sigma`, `max_sigma`. Each is a 2-element vector, giving the minimum and maximum acceptable values for the gaussian sigma. It is important that `min_sigma` is at least 0.5 pixel, estimating gaussian centers when any component of the sigma is significantly smaller than a pixel will lead to poor results anyways.

Definition at line 27 of file `Gauss2DsxyModel.h`.

### 9.19.2 Member Typedef Documentation

#### 9.19.2.1 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 24 of file `ImageFormat2DBase.h`.

#### 9.19.2.2 `using mappel::ImageFormat2DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 25 of file `ImageFormat2DBase.h`.

#### 9.19.2.3 `template<class PixelT> using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>` [inherited]

Shape of the data type for a single image

Definition at line 32 of file `ImageFormat2DBase.h`.

#### 9.19.2.4 `template<class CoordT> using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>` [inherited]

Shape of the data type to store a single image's coordinates

Definition at line 27 of file `ImageFormat2DBase.h`.

#### 9.19.2.5 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 29 of file `ImageFormat2DBase.h`.

**9.19.2.6** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>`  
`[inherited]`

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

**9.19.2.7** `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

**9.19.2.8** `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`  
`[inherited]`

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

**9.19.2.9** `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

**9.19.2.10** `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

**9.19.2.11** `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

**9.19.2.12** `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

**9.19.2.13** `using mappel::Gauss2DsxyModel::StencilVecT = std::vector<Stencil>`

Definition at line 59 of file Gauss2DsxyModel.h.

## 9.19.3 Constructor &amp; Destructor Documentation

9.19.3.1 `mappel::Gauss2DsxyModel::Gauss2DsxyModel ( const ImageSizeT & size, const VecT & min_sigma, const VecT & max_sigma )`

## 9.19.4 Member Function Documentation

9.19.4.1 `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

9.19.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta ( const ParamT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

9.19.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

9.19.4.4 `void mappel::ImageFormat2DBase::check_image_shape ( const ImageT & im ) const` [inherited]

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

9.19.4.5 `void mappel::ImageFormat2DBase::check_image_shape ( const ImageStackT & ims ) const` [inherited]

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

#### 9.19.4.6 void mappel::PointEmitterModel::check\_param\_shape ( const ParamT & *theta* ) const [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

#### 9.19.4.7 void mappel::PointEmitterModel::check\_param\_shape ( const ParamVecT & *theta* ) const [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

#### 9.19.4.8 void mappel::PointEmitterModel::check\_psf\_sigma ( double *psf\_sigma* ) const [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

#### 9.19.4.9 void mappel::PointEmitterModel::check\_psf\_sigma ( const VecT & *psf\_sigma* ) const [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

#### 9.19.4.10 void mappel::ImageFormat2DBase::check\_size ( const ImageSizeT & *size* ) [static],[inherited]

Check the size argument for the model.

Definition at line 60 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::global\_max\_size, and mappel::ImageFormat2DBase::global\_min\_size.

Referenced by mappel::ImageFormat2DBase::ImageFormat2DBase(), and mappel::ImageFormat2DBase::set\_size().

9.19.4.11 `static double mappel::Gauss2DsxyModel::compute_max_sigma_ratio ( const VecT & min_sigma, const VecT & max_sigma ) [static], [protected]`

9.19.4.12 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const [inline], [inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.13 `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const [inline], [inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.14 `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const [inline], [inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

9.19.4.15 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const [inline], [inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.16 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const [inline], [inherited]`

Definition at line 106 of file ImageFormat2DBase.h.

9.19.4.17 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const [inline], [inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.19.4.18** `VecT mappel::Gauss2DsxModel::get_max_sigma ( ) const [inline]`

Definition at line 127 of file Gauss2DsxModel.h.

References `get_max_sigma_ratio()`, and `get_min_sigma()`.

**9.19.4.19** `double mappel::Gauss2DsxModel::get_max_sigma ( IdxT dim ) const [inline]`

Definition at line 131 of file Gauss2DsxModel.h.

References `get_max_sigma_ratio()`, and `get_min_sigma()`.

**9.19.4.20** `double mappel::Gauss2DsxModel::get_max_sigma_ratio ( ) const [inline]`

Definition at line 135 of file Gauss2DsxModel.h.

References `mappel::PointEmitterModel::get_ubound()`.

Referenced by `get_max_sigma()`.

**9.19.4.21** `VecT mappel::Gauss2DsxModel::get_min_sigma ( ) const [inline]`

Definition at line 122 of file Gauss2DsxModel.h.

References `min_sigma`.

Referenced by `get_max_sigma()`.

**9.19.4.22** `double mappel::Gauss2DsxModel::get_min_sigma ( IdxT dim ) const`

**9.19.4.23** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const [inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

**9.19.4.24** `IdxT mappel::PointEmitterModel::get_num_params ( ) const [inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.19.4.25** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_num_pixels ( ) const [inline],[inherited]`

Definition at line 79 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

9.19.4.26 **StringVecT** mappel::PointEmitterModel::get\_param\_names ( ) const [inline],[inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.27 **CompositeDist &** mappel::PointEmitterModel::get\_prior ( ) [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

9.19.4.28 **const CompositeDist &** mappel::PointEmitterModel::get\_prior ( ) const [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.29 **ParallelRngGeneratorT &** mappel::PointEmitterModel::get\_rng\_generator ( ) [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.19.4.30 **ParallelRngManagerT &** mappel::PointEmitterModel::get\_rng\_manager ( ) [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.19.4.31 **const ImageFormat2DBase::ImageSizeT &** mappel::ImageFormat2DBase::get\_size ( ) const [inline],[inherited]

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

9.19.4.32 **ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::get\_size ( IdxT idx ) const [inherited]

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.19.4.33 ImageFormat2DBase::ImageCoordT** `mappel::ImageFormat2DBase::get_size_image_stack ( const ImageStackT & stack ) const` `[inline],[inherited]`

Definition at line 99 of file ImageFormat2DBase.h.

**9.19.4.34 StatsT** `mappel::Gauss2DsxyModel::get_stats ( ) const`

**9.19.4.35 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const** `[inline],[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::ubound`.

Referenced by `get_max_sigma_ratio()`, `mappel::Gauss2DsModel::get_max_sigma_ratio()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

**9.19.4.36 bool** `mappel::PointEmitterModel::has_hyperparam ( const std::string & name ) const` `[inline],[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.19.4.37 Gauss2DsxyModel::Stencil** `mappel::Gauss2DsxyModel::initial_theta_estimate ( const ImageT & im )` `[inline]`

Fast, heuristic estimate of initial theta.

Definition at line 222 of file Gauss2DsxyModel.h.

References `mappel::PointEmitterModel::DefaultSeperableInitEstimator`, and `mappel::PointEmitterModel::make_param()`.

Referenced by `initial_theta_estimate()`.

**9.19.4.38 Gauss2DsxyModel::Stencil** `mappel::Gauss2DsxyModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init )` `[inline]`

Definition at line 229 of file Gauss2DsxyModel.h.

References `mappel::PointEmitterModel::DefaultSeperableInitEstimator`, and `initial_theta_estimate()`.

**9.19.4.39 Stencil** `mappel::Gauss2DsxyModel::initial_theta_estimate ( const ImageT & im, const ParamT & theta_init, const std::string & estimator )`

**9.19.4.40 static CompositeDist** `mappel::Gauss2DsxyModel::make_default_prior ( const ImageSizeT & size, double max_sigma_ratio )` `[static]`

**9.19.4.41 ImageFormat2DBase::ImageT** `mappel::ImageFormat2DBase::make_image ( ) const` `[inline],[inherited]`

Definition at line 85 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.



**9.19.4.42 ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make\_image\_stack ( ImageCoordT *n* ) const**  
 [inline],[inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.19.4.43 PointEmitterModel::ParamT mappel::PointEmitterModel::make\_param ( ) const** [inline],[inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_← estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::← Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

**9.19.4.44 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make\_param ( FillT *fill* ) const**  
 [inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.19.4.45 MatT mappel::PointEmitterModel::make\_param\_mat ( ) const** [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.19.4.46 template<class FillT > MatT mappel::PointEmitterModel::make\_param\_mat ( FillT *fill* ) const** [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.19.4.47 CubeT mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n* ) const** [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.19.4.48 template<class FillT > CubeT mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n*, FillT *fill* ) const**  
 [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.19.4.49** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline]`,  
`[inherited]`

Definition at line 168 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.19.4.50** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.19.4.51** `static CompositeDist mappel::Gauss2DsxyModel::make_prior_beta_position ( const ImageSizeT & size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma )` `[static]`

**9.19.4.52** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_l, double kappa = default_intensity_kappa )` `[static]`, `[inherited]`

Definition at line 98 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.19.4.53** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos )` `[static]`, `[inherited]`

Definition at line 92 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

9.19.4.54 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos ) [static],[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.19.4.55 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma ) [static],[inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.19.4.56 `static CompositeDist mappel::Gauss2DsxyModel::make_prior_normal_position ( const ImageSizeT & size, double sigma_xpos, double sigma_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma ) [static]`

9.19.4.57 `Gauss2DsxyModel::Stencil mappel::Gauss2DsxyModel::make_stencil ( const ParamT & theta, bool compute_derivatives = true ) const [inline]`

Make a new Model::Stencil object at theta.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular theta (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a const [Stencil](#) reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`

#### Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

#### Returns

A new [Stencil](#) object ready to compute with

Definition at line 157 of file Gauss2DsxModel.h.

References `mappel::Gauss2DsxModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.19.4.58** `void mappel::Gauss2DsxModel::pixel_grad ( int i, int j, const Stencil & s, ParamT & pgrad ) const [inline]`

Definition at line 178 of file Gauss2DsxModel.h.

References `mappel::Gauss2DsxModel::Stencil::DX`, `mappel::Gauss2DsxModel::Stencil::DXS`, `mappel::Gauss2DsxModel::Stencil::DY`, `mappel::Gauss2DsxModel::Stencil::DYS`, `mappel::Gauss2DsxModel::Stencil::l()`, `mappel::Gauss2DsxModel::Stencil::X`, and `mappel::Gauss2DsxModel::Stencil::Y`.

**9.19.4.59** `void mappel::Gauss2DsxModel::pixel_grad2 ( int i, int j, const Stencil & s, ParamT & pgrad2 ) const [inline]`

Definition at line 190 of file Gauss2DsxModel.h.

References `mappel::Gauss2DsxModel::Stencil::DXS`, `mappel::Gauss2DsxModel::Stencil::DXS2`, `mappel::Gauss2DsxModel::Stencil::DYS`, `mappel::Gauss2DsxModel::Stencil::DYS2`, `mappel::Gauss2DsxModel::Stencil::l()`, `mappel::Gauss2DsxModel::Stencil::sigmaX()`, `mappel::Gauss2DsxModel::Stencil::sigmaY()`, `mappel::Gauss2DsxModel::Stencil::X`, and `mappel::Gauss2DsxModel::Stencil::Y`.

**9.19.4.60** `void mappel::Gauss2DsxModel::pixel_hess ( int i, int j, const Stencil & s, MatT & hess ) const [inline]`

Definition at line 202 of file Gauss2DsxModel.h.

References `mappel::Gauss2DsxModel::Stencil::DX`, `mappel::Gauss2DsxModel::Stencil::DXS`, `mappel::Gauss2DsxModel::Stencil::DXS2`, `mappel::Gauss2DsxModel::Stencil::DXSX`, `mappel::Gauss2DsxModel::Stencil::DY`, `mappel::Gauss2DsxModel::Stencil::DYS`, `mappel::Gauss2DsxModel::Stencil::DYS2`, `mappel::Gauss2DsxModel::Stencil::DYSY`, `mappel::Gauss2DsxModel::Stencil::l()`, `mappel::Gauss2DsxModel::Stencil::sigmaX()`, `mappel::Gauss2DsxModel::Stencil::sigmaY()`, `mappel::Gauss2DsxModel::Stencil::X`, and `mappel::Gauss2DsxModel::Stencil::Y`.

**9.19.4.61** `void mappel::Gauss2DsxModel::pixel_hess_update ( int i, int j, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess ) const`

**9.19.4.62** `double mappel::Gauss2DsxModel::pixel_model_value ( int i, int j, const Stencil & s ) const [inline]`

Definition at line 171 of file Gauss2DsxModel.h.

References `mappel::Gauss2DsxModel::Stencil::bg()`, `mappel::Gauss2DsxModel::Stencil::l()`, `mappel::Gauss2DsxModel::Stencil::X`, and `mappel::Gauss2DsxModel::Stencil::Y`.

**9.19.4.63** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const [inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.19.4.64** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta )`  
`const [inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.19.4.65** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.19.4.66** `void mappel::Gauss2DsxyModel::sample_mcmc_candidate ( int sample_index, ParamT & candidate_theta, double scale = 1.0 )`

**9.19.4.67** `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.19.4.68** `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const [inline], [inherited]`

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

**9.19.4.69** `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.19.4.70** `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc ) [inline],`  
`[inherited]`

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.19.4.71 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.72 `void mappel::Gauss2DsxyModel::set_hyperparams ( const VecT & hyperparams )`

9.19.4.73 `template<class ImT > void mappel::ImageFormat2DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]`

Definition at line 113 of file ImageFormat2DBase.h.

9.19.4.74 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.19.4.75 `void mappel::Gauss2DsxyModel::set_max_sigma ( const VecT & max_sigma )`

9.19.4.76 `void mappel::Gauss2DsxyModel::set_max_sigma_ratio ( double max_sigma_ratio )`

9.19.4.77 `void mappel::Gauss2DsxyModel::set_min_sigma ( const VecT & min_sigma )`

9.19.4.78 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc ) [inline], [inherited]`

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.79 `void mappel::Gauss2DsxyModel::set_prior ( CompositeDist && prior_ )`

9.19.4.80 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ ) [inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

9.19.4.81 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static], [inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.19.4.82 void mappel::Gauss2DsxyModel::set\_size ( const ImageSizeT & size\_ )

9.19.4.83 void mappel::PointEmitterModel::set\_ubound ( const ParamT & ubound ) [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_max\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.19.4.84 bool mappel::PointEmitterModel::theta\_in\_bounds ( const ParamT & theta ) const [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::make\_stencil(), mappel::Gauss1DsModel::make\_stencil(), mappel::Gauss2DModel::make\_stencil(), make\_stencil(), mappel::Gauss2DsModel::make\_stencil(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.19.4.85 BoolVecT mappel::PointEmitterModel::theta\_stack\_in\_bounds ( const ParamVecT & theta ) const [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::theta\_in\_bounds().

9.19.4.86 void mappel::Gauss2DsxyModel::update\_internal\_1D\_estimators ( ) [protected]

## 9.19.5 Member Data Documentation

9.19.5.1 const double mappel::PointEmitterModel::bounds\_epsilon = 1.0E-6 [static], [inherited]

Distance from the boundary to constrain in bound\_theta and bounded\_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss2DsModel::set\_max\_sigma\_ratio(), and mappel::PointEmitterModel::set\_ubound().

9.19.5.2 const double mappel::PointEmitterModel::default\_alpha\_sigma = 2 [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

**9.19.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

**9.19.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.19.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.19.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.19.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.19.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.



**9.19.5.9** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `initial_theta_estimate()`, and `mappel::Gauss2DsxyModel::initial_theta_estimate()`.

**9.19.5.10** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2 [static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.19.5.11** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size =512 [static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.19.5.12** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1 [static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.19.5.13** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size =3 [static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

#### 9.19.5.14 **ParamT** mappel::PointEmitterModel::lbound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.19.5.15 **double** mappel::Gauss2DsxyModel::mcmc\_candidate\_eta\_sigma [protected]

The standard deviation for the normally distributed perturbation to theta\_sigma in the random walk MCMC sampling

Definition at line 108 of file Gauss2DsxyModel.h.

#### 9.19.5.16 **double** mappel::Gauss2DsxyModel::mcmc\_candidate\_eta\_y [protected]

Std-dev for the normal perturbations to theta\_y under MCMC sampling

Definition at line 107 of file Gauss2DsxyModel.h.

#### 9.19.5.17 **VecT** mappel::Gauss2DsxyModel::min\_sigma [protected]

Gaussian PSF in pixels

Definition at line 113 of file Gauss2DsxyModel.h.

Referenced by get\_min\_sigma().

#### 9.19.5.18 **const ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::num\_dim =2 [static],[inherited]

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::get\_stats().

#### 9.19.5.19 **IdxT** mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

## 9.19.5.20 IdxT mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

## 9.19.5.21 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

## 9.19.5.22 ImageSizeT mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss\_compute\_estimate(), mappel::cgauss\_compute\_estimate\_debug(), mappel::cgauss\_heuristic\_compute\_estimate(), mappel::ImageFormat2DBase::check\_image\_shape(), mappel::Gauss2DModel::Stencil::compute\_derivatives(), mappel::Gauss2DsModel::Stencil::compute\_derivatives(), mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::methods::expected\_information(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsMAP::Gauss2DsMAP(), mappel::Gauss2DsMLE::Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::ImageFormat2DBase::get\_num\_pixels(), mappel::ImageFormat2DBase::get\_size(), mappel::ImageFormat2DBase::get\_stats(), mappel::methods::likelihood::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood::debug::grad\_components(), mappel::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian\_components(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh\_components(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::ImageFormat2DBase::make\_image(), mappel::ImageFormat2DBase::make\_image\_stack(),

mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_prior\_normal\_position(), mappel::methods::model\_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods::likelihood::rllh(), mappel::methods::likelihood::debug::rllh\_components(), mappel::ImageFormat2DBase::set\_size(), mappel::Gauss2DModel::set\_size(), mappel::Gauss2DsModel::set\_size(), mappel::methods::simulate\_image(), mappel::methods::simulate\_image\_from\_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2DsModel::Stencil::Stencil(), mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

#### 9.19.5.23 ParamT mappel::PointEmitterModel::ubound [protected], [inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.19.5.24 Gauss1DsMAP mappel::Gauss2DsxModel::x\_model [protected]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 114 of file Gauss2DsxModel.h.

#### 9.19.5.25 Gauss1DsMAP mappel::Gauss2DsyModel::y\_model [protected]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 115 of file Gauss2DsyModel.h.

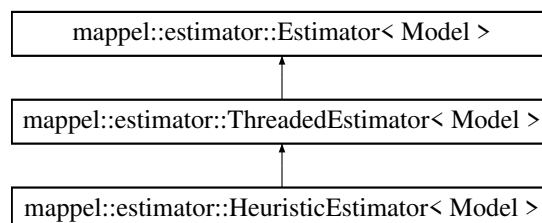
The documentation for this class was generated from the following file:

- [Gauss2DsxModel.h](#)

## 9.20 mappel::estimator::HeuristicEstimator< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::estimator::HeuristicEstimator< Model >:



## Public Member Functions

- [HeuristicEstimator](#) (const Model &model)
- [StatsT get\\_stats](#) ()
- [StatsT get\\_debug\\_stats](#) ()
- [std::string name](#) () const
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [MLEDataStack](#) &mle\_data\_stack) override
- void [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [ProfileLikelihoodData](#) &profile) override
- void [estimate\\_profile\\_bounds\\_parallel](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est) override
- void [estimate\\_profile\\_bounds\\_stack](#) (const [ModelDataStackT](#)< Model > &data, [ProfileBoundsDataStack](#) &bounds\_est\_stack) override
- void [clear\\_stats](#) ()
- const Model & [get\\_model](#) ()
  
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data\_stack, [MLEDataStack](#) &mle\_data\_stack)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, [MLEData](#) &mle\_data)
- void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEDebugData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEDebugData](#) &mle\_data)
  
- double [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed\_idx, const [ParamT](#)< Model > &fixed\_theta\_init, [StencilT](#)< Model > &theta\_max)
  
- void [estimate\\_profile\\_bounds](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est)
- void [estimate\\_profile\\_bounds\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &bounds\_est)
  
- [IdxVecT get\\_exit\\_counts](#) () const

## Protected Member Functions

- void [record\\_exit\\_code](#) ([ExitCode](#) code) override
- virtual void [compute\\_estimate\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEDebugData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- virtual double [compute\\_profile\\_estimate](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, const [IdxVecT](#) &fixed\_idx, [StencilT](#)< Model > &max\_stencil)
- virtual void [compute\\_profile\\_bound](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &est, const [VecT](#) &init\_step, [IdxT](#) param\_idx, [IdxT](#) which\_bound)
- virtual void [compute\\_profile\\_bound\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &est)
- void [record\\_walltime](#) ([ClockT::time\\_point](#) start\_walltime, int [num\\_estimations](#))

## Protected Attributes

- int [max\\_threads](#)
- int [num\\_threads](#)
- std::mutex [mtx](#)
- const Model & [model](#)
- int [num\\_estimations](#) = 0
- double [total\\_walltime](#) = 0.
- [IdxVecT](#) [exit\\_counts](#)

### 9.20.1 Detailed Description

```
template<class Model>
class mappel::estimator::HeuristicEstimator< Model >
```

Definition at line 332 of file estimator.h.

### 9.20.2 Constructor & Destructor Documentation

9.20.2.1 `template<class Model > mappel::estimator::HeuristicEstimator< Model >::HeuristicEstimator ( const Model & model ) [inline]`

Definition at line 335 of file estimator.h.

### 9.20.3 Member Function Documentation

9.20.3.1 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::clear_stats ( ) [virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 570 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::clear\\_stats\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::mtx](#), and [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::clear\\_stats\(\)](#).

9.20.3.2 `template<class Model> void mappel::estimator::Estimator< Model >::compute_estimate_debug ( const ModelDataT< Model > & im, const ParamT< Model > & theta_init, MLEDebugData & mle_debug, StencilT< Model > & mle_stencil )` [protected], [virtual], [inherited]

Virtual estimate\_debug interface

Estimators that produce a sequence of results (e.g. IterativeEstimators) can override this dummy debug implementation.

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 285 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDebugData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max_debug()`.

9.20.3.3 `template<class Model> void mappel::estimator::Estimator< Model >::compute_profile_bound ( const ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 309 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

9.20.3.4 `template<class Model> void mappel::estimator::Estimator< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & est )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 318 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`.

9.20.3.5 `template<class Model> double mappel::estimator::Estimator< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxs, StencilT< Model > & max_stencil )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 300 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

9.20.3.6 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil )`  
[inherited]

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`.

9.20.3.7 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data )` [inherited]

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.



9.20.3.8 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::Estimator< Model >::model`.

9.20.3.9 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at `theta_init`, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.20.3.10 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data )`  
`[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

9.20.3.11 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack )` `[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max_stack()`, and `mappel::estimator::Estimator< Model >::model`.

9.20.3.12 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack )` [override],[virtual],[inherited]

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEDataStack::Ndata`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDataStack::obsI`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDataStack::rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDataStack::theta`.

9.20.3.13 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est )` [inherited]

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

9.20.3.14 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

9.20.3.15 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.

9.20.3.16 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, and `mappel::estimator::ProfileBoundsDataStack::initialize_arrays()`.

mappel::estimator::ProfileBoundsData::mle, mappel::estimator::ProfileBoundsDataStack::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsDataStack::Ndata, mappel::estimator::ProfileBoundsDataStack::Nparams\_est, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::MLEData::obsI, mappel::estimator::MLEDataStack::obsI, mappel::print\_text\_image(), mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_ub, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, mappel::estimator::MLEDataStack::rllh, mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, mappel::estimator::ProfileBoundsDataStack::target\_rllh\_delta, mappel::estimator::MLEData::theta, and mappel::estimator::MLEDataStack::theta.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_stack().

**9.20.3.17** `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), and mappel::methods::objective::rllh().

**9.20.3.18** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override],[virtual],[inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::estimator::ProfileLikelihoodData::fixed\_idx, mappel::estimator::ProfileLikelihoodData::fixed\_values, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileLikelihoodData::Nfixed, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::ProfileLikelihoodData::Nvalues, mappel::print\_text\_image(), mappel::estimator::ProfileLikelihoodData::profile\_likelihood, mappel::estimator::ProfileLikelihoodData::profile\_parameters, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), and mappel::estimator::Estimator< Model >::record\_walltime().

**9.20.3.19** `template<class Model > StatsT mappel::estimator::HeuristicEstimator< Model >::get_debug_stats ( )`  
`[virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 612 of file `estimator_impl.h`.

References `mappel::cgauss_heuristic_compute_estimate()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::Success`, and `mappel::estimator::MLEData::theta`.

**9.20.3.20** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts ( ) const`  
`[inline],[inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.20.3.21** `template<class Model > const Model & mappel::estimator::Estimator< Model >::get_model ( )`  
`[inherited]`

Definition at line 108 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

**9.20.3.22** `template<class Model > StatsT mappel::estimator::HeuristicEstimator< Model >::get_stats ( )`  
`[virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 597 of file `estimator_impl.h`.

References `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, and `mappel::estimator::Estimator< Model >::num_estimations`.

**9.20.3.23** `template<class Model > std::string mappel::estimator::HeuristicEstimator< Model >::name ( ) const`  
`[inline],[virtual]`

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 339 of file `estimator.h`.

9.20.3.24 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code )` [override], [protected], [virtual], [inherited]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::Success`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.20.3.25 `template<class Model> void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` [protected], [inherited]

Definition at line 360 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

## 9.20.4 Member Data Documentation

9.20.4.1 `template<class Model> IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 299 of file `estimator.h`.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

9.20.4.2 `template<class Model> int mappel::estimator::ThreadedEstimator< Model >::max_threads` [protected], [inherited]

Definition at line 324 of file `estimator.h`.

**9.20.4.3** `template<class Model > const Model& mappel::estimator::Estimator< Model >::model` [protected],  
[inherited]

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.20.4.4** `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` [protected],  
[inherited]

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.20.4.5** `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected],  
[inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

**9.20.4.6** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected],  
[inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.



9.20.4.7 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` [protected], [inherited]

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

The documentation for this class was generated from the following files:

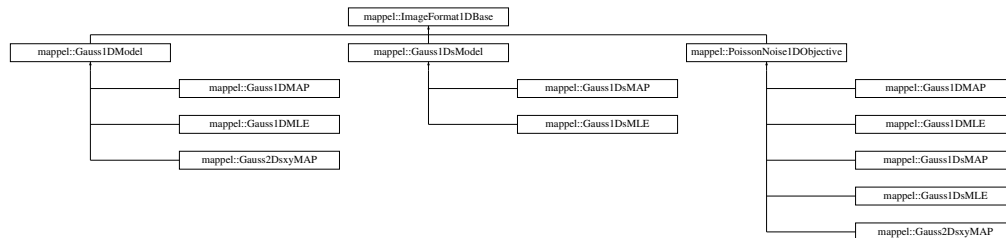
- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.21 mappel::ImageFormat1DBase Class Reference

A virtual base class for 2D image localization objectives.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/ImageFormat1DBase.h>
```

Inheritance diagram for `mappel::ImageFormat1DBase`:



### Public Types

- using `ImageCoordT` = `uint32_t`
- using `ImagePixelT` = `double`
- `template<class CoordT >`  
using `ImageSizeShapeT` = `CoordT`
- `template<class CoordT >`  
using `ImageSizeVecShapeT` = `arma::Col< CoordT >`
- using `ImageSizeT` = `ImageSizeShapeT< ImageCoordT >`
- using `ImageSizeVecT` = `ImageSizeVecShapeT< ImageCoordT >`
- `template<class PixelT >`  
using `ImageShapeT` = `arma::Col< PixelT >`
- `template<class PixelT >`  
using `ImageStackShapeT` = `arma::Mat< PixelT >`
- using `ImageT` = `ImageShapeT< ImagePixelT >`
- using `ImageStackT` = `ImageStackShapeT< ImagePixelT >`

### Public Member Functions

- [StatsT get\\_stats](#) () const
- [ImageT make\\_image](#) () const
- [ImageStackT make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- template<class ImT >  
void [set\\_image\\_in\\_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const ImT &im) const
- [ImageSizeT get\\_size](#) () const
- [ImageCoordT get\\_size](#) ([IdxT](#) idx) const
- [ImageCoordT get\\_num\\_pixels](#) () const
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- void [set\\_size](#) (const arma::Col< [ImageCoordT](#) > &sz)
- void [check\\_image\\_shape](#) (const [ImageT](#) &im) const  
*Check the shape of a single images is correct for model size.*
- void [check\\_image\\_shape](#) (const [ImageStackT](#) &ims) const  
*Check the shape of a stack of images is correct for model size.*

### Static Public Member Functions

- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)  
*Check the size argument for the model.*

### Static Public Attributes

- static const [ImageCoordT](#) num\_dim = 1
- static const [ImageCoordT](#) global\_min\_size = 3
- static const [ImageCoordT](#) global\_max\_size = 512

### Protected Member Functions

- [ImageFormat1DBase](#) ()=default
- [ImageFormat1DBase](#) ([ImageSizeT](#) size\_)

### Protected Attributes

- [ImageSizeT](#) size

#### 9.21.1 Detailed Description

A virtual base class for 2D image localization objectives.

This class should be inherited virtually by both the model and the objective so that the common image information and functions are available in both Model and Objective classes hierarchies

Definition at line 23 of file [ImageFormat1DBase.h](#).

### 9.21.2 Member Typedef Documentation

#### 9.21.2.1 using mappel::ImageFormat1DBase::ImageCoordT = uint32\_t

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

#### 9.21.2.2 using mappel::ImageFormat1DBase::ImagePixelT = double

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

#### 9.21.2.3 template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT>

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

#### 9.21.2.4 template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

#### 9.21.2.5 using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

#### 9.21.2.6 template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT>

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

#### 9.21.2.7 using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

#### 9.21.2.8 template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

#### 9.21.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

#### 9.21.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

### 9.21.3 Constructor & Destructor Documentation

#### 9.21.3.1 `mappel::ImageFormat1DBase::ImageFormat1DBase ( )` `[protected]`, `[default]`

#### 9.21.3.2 `ImageFormat1DBase::ImageFormat1DBase ( ImageSizeT size_ )` `[explicit]`, `[protected]`

Definition at line 13 of file ImageFormat1DBase.cpp.

References `check_size()`.

### 9.21.4 Member Function Documentation

#### 9.21.4.1 `void ImageFormat1DBase::check_image_shape ( const ImageT & im ) const`

Check the shape of a single images is correct for model size.

Definition at line 59 of file ImageFormat1DBase.cpp.

References `size`.

#### 9.21.4.2 `void ImageFormat1DBase::check_image_shape ( const ImageStackT & ims ) const`

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file ImageFormat1DBase.cpp.

References `size`.

#### 9.21.4.3 `void ImageFormat1DBase::check_size ( const ImageSizeT & size_ )` `[static]`

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References `global_max_size`, and `global_min_size`.

Referenced by `ImageFormat1DBase()`, and `set_size()`.

**9.21.4.4** `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const [inline]`

Definition at line 108 of file ImageFormat1DBase.h.

**9.21.4.5** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_num_pixels ( ) const [inline]`

Definition at line 82 of file ImageFormat1DBase.h.

References size.

Referenced by get\_stats().

**9.21.4.6** `ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get_size ( ) const [inline]`

Definition at line 71 of file ImageFormat1DBase.h.

References size.

Referenced by get\_stats().

**9.21.4.7** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size ( IdxT idx ) const`

Definition at line 20 of file ImageFormat1DBase.cpp.

References size.

**9.21.4.8** `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack ( const ImageStackT & stack ) const [inline]`

Definition at line 101 of file ImageFormat1DBase.h.

**9.21.4.9** `StatsT ImageFormat1DBase::get_stats ( ) const`

Definition at line 81 of file ImageFormat1DBase.cpp.

References get\_num\_pixels(), get\_size(), and num\_dim.

Referenced by mappel::Gauss1DModel::get\_stats(), and mappel::Gauss1DsModel::get\_stats().

**9.21.4.10** `ImageFormat1DBase::ImageT ImageFormat1DBase::make_image ( ) const [inline]`

Definition at line 87 of file ImageFormat1DBase.h.

References size.

**9.21.4.11** `ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack ( ImageCoordT n ) const`  
`[inline]`

Definition at line 94 of file ImageFormat1DBase.h.

References `size`.

**9.21.4.12** `template<class ImT > void ImageFormat1DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n,  
const ImT & im ) const`

Definition at line 115 of file ImageFormat1DBase.h.

**9.21.4.13** `void ImageFormat1DBase::set_size ( const ImageSizeT & size_ )`

Definition at line 30 of file ImageFormat1DBase.cpp.

References `check_size()`, and `size`.

Referenced by `set_size()`, `mappel::Gauss2DModel::set_size()`, and `mappel::Gauss2DsModel::set_size()`.

**9.21.4.14** `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz )` `[inline]`

Definition at line 75 of file ImageFormat1DBase.h.

References `set_size()`.

## 9.21.5 Member Data Documentation

**9.21.5.1** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file ImageFormat1DBase.h.

Referenced by `check_size()`.

**9.21.5.2** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static]`

Minimum size along any dimension of the image.

Definition at line 39 of file ImageFormat1DBase.h.

Referenced by `check_size()`.

**9.21.5.3** `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by `get_stats()`.

## 9.21.5.4 ImageSizeT mappel::ImageFormat1DBase::size [protected]

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by check\_image\_shape(), mappel::Gauss1DsModel::Stencil::compute\_derivatives(), mappel::Gauss1DModel::Stencil::compute\_derivatives(), get\_num\_pixels(), get\_size(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), make\_image(), make\_image\_stack(), set\_size(), mappel::Gauss1DsModel::Stencil::Stencil(), and mappel::Gauss1DModel::Stencil::Stencil().

The documentation for this class was generated from the following files:

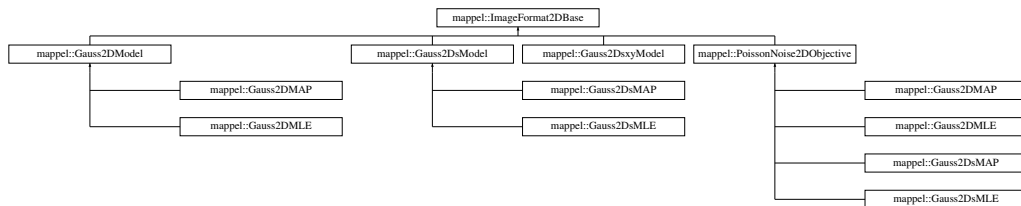
- [ImageFormat1DBase.h](#)
- [ImageFormat1DBase.cpp](#)

## 9.22 mappel::ImageFormat2DBase Class Reference

A virtual base class for 2D image localization objectives.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/ImageFormat2DBase.h>
```

Inheritance diagram for mappel::ImageFormat2DBase:



## Public Types

- using [ImageCoordT](#) = uint32\_t
- using [ImagePixelT](#) = double
- template<class CoordT >  
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >  
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >  
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >  
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >

### Public Member Functions

- [StatsT get\\_stats](#) () const
- [ImageT make\\_image](#) () const
- [ImageStackT make\\_image\\_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get\\_size\\_image\\_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT get\\_image\\_from\\_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- template<class [ImT](#) >  
void [set\\_image\\_in\\_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const [ImT](#) &im) const
- const [ImageSizeT](#) & [get\\_size](#) () const
- [ImageCoordT get\\_size](#) ([IdxT](#) idx) const
- [ImageCoordT get\\_num\\_pixels](#) () const
- void [set\\_size](#) (const [ImageSizeT](#) &size\_)
- void [check\\_image\\_shape](#) (const [ImageT](#) &im) const  
*Check the shape of a single images is correct for model size.*
- void [check\\_image\\_shape](#) (const [ImageStackT](#) &ims) const  
*Check the shape of a stack of images is correct for model size.*

### Static Public Member Functions

- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)  
*Check the size argument for the model.*

### Static Public Attributes

- static const [ImageCoordT](#) [num\\_dim](#) =2
- static const [ImageCoordT](#) [global\\_min\\_size](#) =3
- static const [ImageCoordT](#) [global\\_max\\_size](#) =512

### Protected Member Functions

- [ImageFormat2DBase](#) ()=default
- [ImageFormat2DBase](#) (const [ImageSizeT](#) &size)
- [ImageFormat2DBase](#) (const [ImageFormat2DBase](#) &)
- [ImageFormat2DBase](#) ([ImageFormat2DBase](#) &&)
- [ImageFormat2DBase](#) & [operator=](#) (const [ImageFormat2DBase](#) &)
- [ImageFormat2DBase](#) & [operator=](#) ([ImageFormat2DBase](#) &&)

### Protected Attributes

- [ImageSizeT](#) [size](#)



### 9.22.1 Detailed Description

A virtual base class for 2D image localization objectives.

This class should be inherited virtually by both the model and the objective so that the common image information and functions are available in both Model and Objective classes hierarchies

Definition at line 22 of file ImageFormat2DBase.h.

### 9.22.2 Member Typedef Documentation

#### 9.22.2.1 using mappel::ImageFormat2DBase::ImageCoordT = uint32\_t

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

#### 9.22.2.2 using mappel::ImageFormat2DBase::ImagePixelT = double

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

#### 9.22.2.3 template<class PixelT > using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>

Shape of the data type for a single image

Definition at line 32 of file ImageFormat2DBase.h.

#### 9.22.2.4 template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>

Shape of the data type to store a single image's coordinates

Definition at line 27 of file ImageFormat2DBase.h.

#### 9.22.2.5 using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

#### 9.22.2.6 template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

#### 9.22.2.7 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

#### 9.22.2.8 `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

#### 9.22.2.9 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

#### 9.22.2.10 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

### 9.22.3 Constructor & Destructor Documentation

#### 9.22.3.1 `mappel::ImageFormat2DBase::ImageFormat2DBase ( ) [protected],[default]`

#### 9.22.3.2 `mappel::ImageFormat2DBase::ImageFormat2DBase ( const ImageSizeT & size ) [explicit],[protected]`

Definition at line 13 of file ImageFormat2DBase.cpp.

References `check_size()`.

#### 9.22.3.3 `mappel::ImageFormat2DBase::ImageFormat2DBase ( const ImageFormat2DBase & o ) [protected]`

Definition at line 19 of file ImageFormat2DBase.cpp.

#### 9.22.3.4 `mappel::ImageFormat2DBase::ImageFormat2DBase ( ImageFormat2DBase && o ) [protected]`

Definition at line 23 of file ImageFormat2DBase.cpp.

### 9.22.4 Member Function Documentation

#### 9.22.4.1 void mappel::ImageFormat2DBase::check\_image\_shape ( const ImageT & *im* ) const

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References size.

#### 9.22.4.2 void mappel::ImageFormat2DBase::check\_image\_shape ( const ImageStackT & *ims* ) const

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References size.

#### 9.22.4.3 void mappel::ImageFormat2DBase::check\_size ( const ImageSizeT & *size\_* ) [static]

Check the size argument for the model.

Definition at line 60 of file ImageFormat2DBase.cpp.

References global\_max\_size, and global\_min\_size.

Referenced by ImageFormat2DBase(), and set\_size().

#### 9.22.4.4 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get\_image\_from\_stack ( const ImageStackT & *stack*, ImageCoordT *n* ) const [inline]

Definition at line 106 of file ImageFormat2DBase.h.

#### 9.22.4.5 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get\_num\_pixels ( ) const [inline]

Definition at line 79 of file ImageFormat2DBase.h.

References size.

Referenced by get\_stats().

#### 9.22.4.6 const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get\_size ( ) const [inline]

Definition at line 74 of file ImageFormat2DBase.h.

References size.

#### 9.22.4.7 **ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::get\_size ( **IdxT idx** ) const

Definition at line 41 of file ImageFormat2DBase.cpp.

References size.

#### 9.22.4.8 **ImageFormat2DBase::ImageCoordT** mappel::ImageFormat2DBase::get\_size\_image\_stack ( const **ImageStackT & stack** ) const `[inline]`

Definition at line 99 of file ImageFormat2DBase.h.

#### 9.22.4.9 **StatsT** mappel::ImageFormat2DBase::get\_stats ( ) const

Definition at line 103 of file ImageFormat2DBase.cpp.

References get\_num\_pixels(), num\_dim, and size.

Referenced by mappel::Gauss2DModel::get\_stats(), and mappel::Gauss2DsModel::get\_stats().

#### 9.22.4.10 **ImageFormat2DBase::ImageT** mappel::ImageFormat2DBase::make\_image ( ) const `[inline]`

Definition at line 85 of file ImageFormat2DBase.h.

References size.

#### 9.22.4.11 **ImageFormat2DBase::ImageStackT** mappel::ImageFormat2DBase::make\_image\_stack ( **ImageCoordT n** ) const `[inline]`

Definition at line 92 of file ImageFormat2DBase.h.

References size.

#### 9.22.4.12 **ImageFormat2DBase & mappel::ImageFormat2DBase::operator=** ( const **ImageFormat2DBase & o** ) `[protected]`

Definition at line 27 of file ImageFormat2DBase.cpp.

References size.

Referenced by mappel::Gauss2DsMAP::operator=(), mappel::Gauss2DsMLE::operator=(), mappel::Gauss2DMAP↔::operator=(), and mappel::Gauss2DMLE::operator=().

#### 9.22.4.13 **ImageFormat2DBase & mappel::ImageFormat2DBase::operator=** ( **ImageFormat2DBase && o** ) `[protected]`

Definition at line 33 of file ImageFormat2DBase.cpp.

References size.

9.22.4.14 `template<class ImT > void mappel::ImageFormat2DBase::set_image_in_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const`

Definition at line 113 of file ImageFormat2DBase.h.

9.22.4.15 `void mappel::ImageFormat2DBase::set_size ( const ImageSizeT & size_ )`

Definition at line 51 of file ImageFormat2DBase.cpp.

References `check_size()`, and `size`.

Referenced by `mappel::Gauss2DModel::set_size()`, and `mappel::Gauss2DsModel::set_size()`.

## 9.22.5 Member Data Documentation

9.22.5.1 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size=512` `[static]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `check_size()`.

9.22.5.2 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size=3` `[static]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `check_size()`.

9.22.5.3 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim=2` `[static]`

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by `get_stats()`.

#### 9.22.5.4 `ImageSizeT` `mappel::ImageFormat2DBase::size` [protected]

Number of pixels in X dimension for 1D image

Definition at line 67 of file `ImageFormat2DBase.h`.

Referenced by `mappel::cgauss_compute_estimate()`, `mappel::cgauss_compute_estimate_debug()`, `mappel::cgauss_`  
`_heuristic_compute_estimate()`, `check_image_shape()`, `mappel::Gauss2DModel::Stencil::compute_derivatives()`,  
`mappel::Gauss2DsModel::Stencil::compute_derivatives()`, `mappel::Gauss2DModel::debug_internal_sum_model_`  
`y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::methods::expected_information()`, `mappel::`  
`Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsMAP::Gauss2DsMAP()`, `mappel::Gauss2DsMLE::Gauss2Ds`  
`MLE()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `get_num_pixels()`, `get_size()`, `get_stats()`, `mappel::methods`  
`::likelihood::grad()`, `mappel::methods::likelihood::grad2()`, `mappel::methods::likelihood::debug::grad_components()`,  
`mappel::methods::likelihood::hessian()`, `mappel::methods::likelihood::debug::hessian_components()`, `mappel::`  
`Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::methods`  
`::likelihood::llh()`, `mappel::methods::likelihood::debug::llh_components()`, `mappel::Gauss2DModel::make_default_`  
`prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_`  
`_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `make_image()`,  
`make_image_stack()`, `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DsModel::make_`  
`_internal_1Dsum_estimator()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel`  
`::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss2DsModel`  
`::make_prior_normal_position()`, `mappel::methods::model_image()`, `operator=()`, `mappel::methods::likelihood::rllh()`,  
`mappel::methods::likelihood::debug::rllh_components()`, `set_size()`, `mappel::Gauss2DModel::set_size()`, `mappel::`  
`Gauss2DsModel::set_size()`, `mappel::methods::simulate_image()`, `mappel::methods::simulate_image_from_model()`,  
`mappel::Gauss2DModel::Stencil::Stencil()`, `mappel::Gauss2DsModel::Stencil::Stencil()`, `mappel::Gauss2DModel`  
`::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

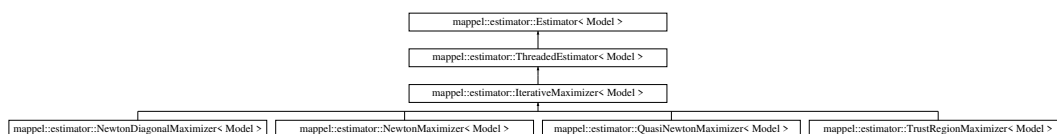
The documentation for this class was generated from the following files:

- [ImageFormat2DBase.h](#)
- [ImageFormat2DBase.cpp](#)

## 9.23 `mappel::estimator::IterativeMaximizer< Model >` Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::estimator::IterativeMaximizer< Model >`:



### Classes

- class [MaximizerData](#)

## Public Member Functions

- [IterativeMaximizer](#) (const Model &model, int max\_iterations=DefaultIterations)
  - double [mean\\_iterations](#) ()
  - double [mean\\_backtracks](#) ()
  - double [mean\\_fun\\_evals](#) ()
  - double [mean\\_der\\_evals](#) ()
  - StatsT [get\\_stats](#) ()
  - StatsT [get\\_debug\\_stats](#) ()
  - void [clear\\_stats](#) ()
  - int [get\\_total\\_iterations](#) () const
  - int [get\\_total\\_backtracks](#) () const
  - int [get\\_total\\_fun\\_evals](#) () const
  - int [get\\_total\\_der\\_evals](#) () const
  - void [local\\_maximize](#) (const [ModelDataT](#)< Model > &im, [StencilT](#)< Model > &stencil, [MLEData](#) &data)
 

*Perform a local maximization to finish off a simulated annealing run.*
  - void [local\\_maximize](#) (const [ModelDataT](#)< Model > &im, [StencilT](#)< Model > &stencil, [MLEDebugData](#) &debug←\_data)
  - void [local\\_profile\\_maximize](#) (const [ModelDataT](#)< Model > &im, const [IdxVecT](#) &fixed\_param\_idxes, [StencilT](#)< Model > &stencil, [MLEDebugData](#) &mle)
  - void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init←\_stack, [MLEDataStack](#) &mle\_data\_stack) override
  - void [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [ProfileLikelihoodData](#) &profile) override
  - void [estimate\\_profile\\_bounds\\_parallel](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est) override
  - void [estimate\\_profile\\_bounds\\_stack](#) (const [ModelDataStackT](#)< Model > &data, [ProfileBoundsDataStack](#) &bounds\_est\_stack) override
  - virtual std::string [name](#) () const =0
  - const Model & [get\\_model](#) ()
- 
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data\_stack, [MLEDataStack](#) &mle\_data\_stack)
  - void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle←\_data, [StencilT](#)< Model > &mle\_stencil)
  - void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle←\_data)
  - void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, [MLEData](#) &mle\_data)
  - void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLE←\\_DebugData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
  - void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLE←\\_DebugData](#) &mle\_data)
- 
- double [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed\_idxes, const [ParamT](#)< Model > &fixed\_theta\_init, [StencilT](#)< Model > &theta\_max)
- 
- void [estimate\\_profile\\_bounds](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est)
  - void [estimate\\_profile\\_bounds\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &bounds←\_est)
- 
- [IdxVecT](#) [get\\_exit\\_counts](#) () const

## Static Public Attributes

- static const int `DefaultIterations` =100

## Protected Member Functions

- void `record_run_statistics` (const `MaximizerData` &data)
- void `compute_estimate` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil) override
- void `compute_estimate_debug` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil) override
- double `compute_profile_estimate` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, const `IdxVecT` &fixed\_idx, `StencilT`< Model > &theta\_max) override
- void `compute_profile_bound` (const `ModelDataT`< Model > &data, `ProfileBoundsData` &est, const `VecT` &init\_← step, `IdxT` param\_idx, `IdxT` which\_bound) override
- void `compute_profile_bound_debug` (const `ModelDataT`< Model > &data, `ProfileBoundsDebugData` &bounds) override
- bool `backtrack` (`MaximizerData` &data)
- bool `profile_bound_backtrack` (`MaximizerData` &data, `IdxT` fixed\_idx, double target\_rllh, double old\_fval, const `VecT` &fgrad)
- virtual void `maximize` (`MaximizerData` &data)=0
- virtual void `solve_profile_bound` (`MaximizerData` &data, `MLEDData` &mle, double llh\_delta, `IdxT` fixed\_idx, `IdxT` which\_bound)
- bool `convergence_test_grad_ratio` (const `VecT` &grad, double fval)
- bool `convergence_test_step_size` (const `VecT` &new\_theta, const `VecT` &old\_theta)
- void `record_exit_code` (`ExitCode` code) override
- void `record_walltime` (`ClockT`::time\_point start\_walltime, int num\_estimations)

## Protected Attributes

- int `max_iterations`
- int `total_iterations` = 0
- int `total_backtracks` = 0
- int `total_fun_evals` = 0
- int `total_der_evals` = 0
- `IdxVecT` `last_backtrack_idx`

*Debugging: Stores last set of backtrack\_idx when data.save\_seq==true.*

- int `max_threads`
- int `num_threads`
- std::mutex `mtx`
- const Model & `model`
- int `num_estimations` = 0
- double `total_walltime` = 0.
- `IdxVecT` `exit_counts`



## Static Protected Attributes

- static const double [min\\_eigenvalue\\_correction\\_delta](#) = 1e-3  
*Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.*
- static const double [convergence\\_min\\_function\\_change\\_ratio](#) = 1.0e-9  
*Convergence criteria: tolerance for function-value change.*
- static const double [convergence\\_min\\_step\\_size\\_ratio](#) = 1.0e-9  
*Convergence criteria: tolerance of relative step size.*
- static const double [backtrack\\_min\\_ratio](#) = 0.05
- static const double [backtrack\\_max\\_ratio](#) = 0.50
- static const double [backtrack\\_min\\_linear\\_step\\_ratio](#) = 1e-3
- static const int [max\\_backtracks](#) = 8
- static const double [min\\_profile\\_bound\\_residual](#) = 1e-4  
*Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.*

## 9.23.1 Detailed Description

```
template<class Model>
class mappel::estimator::IterativeMaximizer< Model >
```

Definition at line 424 of file estimator.h.

## 9.23.2 Constructor &amp; Destructor Documentation

9.23.2.1 `template<class Model > mappel::estimator::IterativeMaximizer< Model >::IterativeMaximizer ( const Model & model, int max_iterations = DefaultIterations )`

Definition at line 732 of file estimator\_impl.h.

## 9.23.3 Member Function Documentation

9.23.3.1 `template<class Model > bool mappel::estimator::IterativeMaximizer< Model >::backtrack ( MaximizerData & data ) [protected]`

Definition at line 870 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio`, `mappel::clamp()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::MaxBacktracks`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::StepSize`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.23.3.2 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::clear_stats ( ) [virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model>](#).

Definition at line 848 of file `estimator_impl.h`.

References `mappel::estimator::ThreadedEstimator< Model>::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model>::mtx`, `mappel::estimator::IterativeMaximizer< Model>::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model>::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model>::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model>::total_iterations`.

9.23.3.3 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::compute_estimate ( const ModelDataT< Model> & data, const ParamT< Model> & theta_init, MLEData & mle_data, StencilT< Model> & mle_stencil ) [override], [protected], [virtual]`

Implements [mappel::estimator::Estimator< Model>](#).

Definition at line 1043 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.23.3.4 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug ( const ModelDataT< Model> & im, const ParamT< Model> & theta_init, MLEDebugData & mle_debug, StencilT< Model> & mle_stencil ) [override], [protected], [virtual]`

Virtual `estimate_debug` interface

Estimators that produce a sequence of results (e.g. `IterativeEstimators`) can override this dummy debug implementation.

Reimplemented from [mappel::estimator::Estimator< Model>](#).

Definition at line 1057 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEDebugData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.23.3.5 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound ( const ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound )` [override], [protected], [virtual]

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1088 of file estimator\_impl.h.

References [mappel::estimator::ProfileBoundsData::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsData::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsData::profile\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_ub](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::solve\\_profile\\_bound\(\)](#), [mappel::estimator::ProfileBoundsData::target\\_rllh\\_delta](#), and [mappel::estimator::MLEData::theta](#).

9.23.3.6 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds )` [override], [protected], [virtual]

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1114 of file estimator\_impl.h.

References [mappel::estimator::ProfileBoundsDebugData::estimated\\_idx](#), [mappel::estimator::ProfileBoundsDebugData::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsDebugData::Nseq\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::Nseq\\_ub](#), [mappel::estimator::MLEData::obs](#), [mappel::estimator::ProfileBoundsDebugData::profile\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::profile\\_ub](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_ub](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_ub\\_rllh](#), [mappel::estimator::IterativeMaximizer< Model >::solve\\_profile\\_bound\(\)](#), [mappel::estimator::subroutine::solve\\_profile\\_initial\\_step\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step](#), [mappel::estimator::ProfileBoundsDebugData::target\\_rllh\\_delta](#), and [mappel::estimator::MLEData::theta](#).

9.23.3.7 `template<class Model> double mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxes, StencilT< Model > & theta_max )` [override], [protected], [virtual]

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1074 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_fixed\\_parameters\(\)](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

**9.23.3.8** `template<class Model> bool mappel::estimator::IterativeMaximizer< Model>::convergence_test_grad_ratio ( const VecT & grad, double fval ) [protected]`

Definition at line 1015 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::convergence_min_function_change_ratio`, `mappel::estimator::GradRatio`, `mappel::norm_sq()`, `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::square()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.23.3.9** `template<class Model> bool mappel::estimator::IterativeMaximizer< Model>::convergence_test_step_size ( const VecT & new_theta, const VecT & old_theta ) [protected]`

Definition at line 1027 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::convergence_min_step_size_ratio`, `mappel::norm_sq()`, `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::estimator::StepSize`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.23.3.10** `template<class Model> void mappel::estimator::Estimator< Model>::estimate_max ( const ModelDataT< Model> & data, const ParamT< Model> & theta_init, MLEData & mle_data, StencilT< Model> & mle_stencil ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model>::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model>::record_exit_code()`, `mappel::estimator::Estimator< Model>::record_walltime()`, `mappel::estimator::MLEData::rllh`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::Estimator< Model>::estimate_max()`.

9.23.3.11 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.

9.23.3.12 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::Estimator< Model >::model`.

9.23.3.13 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at theta\_init, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator.

If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

**9.23.3.14** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

**9.23.3.15** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::estimate\\_max\\_stack\(\)](#), and [mappel::estimator::Estimator< Model >::model](#).

**9.23.3.16** `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack )` `[override]`, `[virtual]`, `[inherited]`

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with *theta\_init*, *theta\_init* is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of *theta\_init* will not be modified in the initialization process.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::MLEDataStack::Ndata](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::MLEData::obsI](#), [mappel::estimator::MLEDataStack::obsI](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), [mappel::estimator::MLEData::rllh](#), [mappel::estimator::MLEDataStack::rllh](#), [mappel::estimator::MLEData::theta](#), and [mappel::estimator::MLEDataStack::theta](#).

**9.23.3.17** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est )` `[inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

**9.23.3.18** `template<class Model> void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model> & data, ProfileBoundsDebugData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

**9.23.3.19** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model> & data, ProfileBoundsData & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.



9.23.3.20 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_bound\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::ProfileBoundsData::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsDataStack::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsData::initialize\\_arrays\(\)](#), [mappel::estimator::ProfileBoundsDataStack::initialize\\_arrays\(\)](#), [mappel::estimator::ProfileBoundsData::mle](#), [mappel::estimator::ProfileBoundsDataStack::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsDataStack::Ndata](#), [mappel::estimator::ProfileBoundsDataStack::Nparams\\_est](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::MLEData::obs](#), [mappel::estimator::MLEDataStack::obs](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ProfileBoundsData::profile\\_lb](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub\\_rllh](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_ub\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_ub](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_ub](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), [mappel::estimator::MLEData::rllh](#), [mappel::estimator::MLEDataStack::rllh](#), [mappel::estimator::subroutine::solve\\_profile\\_initial\\_step\(\)](#), [mappel::estimator::ProfileBoundsData::target\\_rllh\\_delta](#), [mappel::estimator::ProfileBoundsDataStack::target\\_rllh\\_delta](#), [mappel::estimator::MLEData::theta](#), and [mappel::estimator::MLEDataStack::theta](#).

Referenced by [mappel::methods::openmp::error\\_bounds\\_profile\\_likelihood\\_stack\(\)](#).

9.23.3.21 `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::Estimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

9.23.3.22 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override], [virtual], [inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::ProfileLikelihoodData::fixed\\_idx](#)s, [mappel::estimator::ProfileLikelihoodData::fixed\\_values](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileLikelihoodData::Nfixed](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::ProfileLikelihoodData::Nvalues](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ProfileLikelihoodData::profile\\_likelihood](#), [mappel::estimator::ProfileLikelihoodData::profile\\_parameters](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#).

**9.23.3.23** `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_debug_stats ( )`  
`[virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 832 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx`s.

**9.23.3.24** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts ( ) const`  
`[inline], [inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.23.3.25** `template<class Model > const Model & mappel::estimator::Estimator< Model >::get_model ( )`  
`[inherited]`

Definition at line 108 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

**9.23.3.26** `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_stats ( )`  
`[virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 811 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::Estimator< Model >::num_estimations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`, `mappel::methods::debug::error_bounds_profile_likelihood_debug()`, `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`, `mappel::methods::openmp::error_bounds_profile_likelihood_stack()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`.

9.23.3.27 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::get_total_backtracks ( ) const`  
`[inline]`

Definition at line 441 of file estimator.h.

9.23.3.28 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals ( ) const`  
`[inline]`

Definition at line 443 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.23.3.29 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals ( ) const`  
`[inline]`

Definition at line 442 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.23.3.30 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::get_total_iterations ( ) const`  
`[inline]`

Definition at line 440 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.23.3.31 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::local_maximize ( const`  
`ModelDataT< Model > & im, StencilT< Model > & stencil, MLEData & data )`

Perform a local maximization to finish off a simulated annealing run.

Definition at line 1145 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.23.3.32 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::local_maximize ( const`  
`ModelDataT< Model > & im, StencilT< Model > & stencil, MLEDebugData & debug_data )`

Definition at line 1158 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::MLEDebugData::theta`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

9.23.3.33 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize ( const ModelDataT< Model > & im, const IdxVecT & fixed_param_idx, StencilT< Model > & stencil, MLEDebugData & mle )`

Definition at line 1173 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::subroutine::bound_step()`, `mappel::clamp()`, `mappel::estimator::subroutine::compute_bound_scaling_vec()`, `mappel::estimator::subroutine::compute_cauchy_point()`, `mappel::estimator::subroutine::compute_initial_trust_radius()`, `mappel::estimator::subroutine::compute_quadratic_model_value()`, `mappel::estimator::subroutine::compute_scaled_problem()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::fixed_idx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::free_idx`, `mappel::estimator::Estimator< Model >::get_exit_counts()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_iterations()`, `mappel::methods::objective::grad()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::methods::objective::grad2()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters()`, `mappel::methods::objective::hessian()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::is_positive_definite()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::MaxIter`, `mappel::estimator::IterativeMaximizer< Model >::min_profile_bound_residual`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters()`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s0`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s1`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::estimator::subroutine::solve_TR_subproblem()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::MLEData::theta`, `mappel::estimator::MLEDebugData::theta`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_iterations`, and `mappel::estimator::TrustRegionRadius`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.23.3.34 `template<class Model > virtual void mappel::estimator::IterativeMaximizer< Model >::maximize ( MaximizerData & data ) [protected],[pure virtual]`

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.23.3.35 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_backtracks ( )`

9.23.3.36 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_der_evals ( )`

9.23.3.37 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_fun_evals ( )`

9.23.3.38 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_iterations ( )`

9.23.3.39 `template<class Model > virtual std::string mappel::estimator::Estimator< Model >::name ( ) const [pure virtual],[inherited]`

Implemented in [mappel::estimator::TrustRegionMaximizer< Model >](#), [mappel::estimator::QuasiNewtonMaximizer< Model >](#), [mappel::estimator::NewtonMaximizer< Model >](#), [mappel::estimator::NewtonDiagonalMaximizer< Model >](#), [mappel::estimator::SimulatedAnnealingMaximizer< Model >](#), [mappel::estimator::CGaussMLE< Model >](#), [mappel::estimator::CGaussHeuristicEstimator< Model >](#), and [mappel::estimator::HeuristicEstimator< Model >](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::solve\\_profile\\_bound\(\)](#).

9.23.3.40 `template<class Model > bool mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack ( MaximizerData & data, IdxT fixed_idx, double target_rllh, double old_fval, const VecT & fgrad ) [protected]`

Definition at line 943 of file `estimator_impl.h`.

References [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_max\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_linear\\_step\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_ratio](#), [mappel::clamp\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_step\\_size\\_ratio](#), [mappel::methods::objective::grad\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im](#), [mappel::estimator::IterativeMaximizer< Model >::max\\_backtracks](#), [mappel::estimator::MaxBacktracks](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_backtrack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_iteration\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore\\_stencil\(\)](#), [mappel::methods::objective::rllh\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save\\_stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved\\_theta\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_stencil\(\)](#), [mappel::square\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step](#), [mappel::estimator::StepSize](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.23.3.41 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code ) [override],[protected],[virtual],[inherited]`

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::exit\\_counts](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ThreadedEstimator< Model >::mtx](#), [mappel::methods::observed\\_information\(\)](#), [mappel::estimator::MLEData::obsI](#), [mappel::estimator::MLEData::rllh](#), [mappel::methods::objective::rllh\(\)](#), [mappel::estimator::Success](#), and [mappel::estimator::MLEData::theta](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_test\\_grad\\_ratio\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_test\\_step\\_size\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_max\\_stack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_parallel\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_stack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_max\(\)](#), [mappel::estimator::HeuristicEstimator< Model >::get\\_debug\\_stats\(\)](#), [mappel::estimator::CGaussMLE< Model >::get\\_debug\\_stats\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#), and [mappel::estimator::IterativeMaximizer< Model >::profile\\_bound\\_backtrack\(\)](#).

**9.23.3.42** `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::record_run_statistics ( const MaximizerData & data ) [protected]`

Definition at line 859 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_backtrack_idxes()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::has_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model>::last_backtrack_idxes`, `mappel::estimator::ThreadedEstimator< Model>::mtx`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::nBacktracks`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::nIterations`, `mappel::estimator::IterativeMaximizer< Model>::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model>::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model>::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model>::total_iterations`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.23.3.43** `template<class Model> void mappel::estimator::Estimator< Model>::record_walltime ( ClockT::time_point start_walltime, int num_estimations ) [protected], [inherited]`

Definition at line 360 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model>::num_estimations`, and `mappel::estimator::Estimator< Model>::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model>::estimate_max()`, `mappel::estimator::Estimator< Model>::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::estimator::Estimator< Model>::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model>::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model>::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_max()`.

**9.23.3.44** `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::solve_profile_bound ( MaximizerData & data, MLEData & mle, double llh_delta, IdxT fixed_idx, IdxT which_bound ) [protected], [virtual]`

Definition at line 1137 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model>::name()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound()`, and `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound_debug()`.

## 9.23.4 Member Data Documentation

**9.23.4.1** `template<class Model> const double mappel::estimator::IterativeMaximizer< Model>::backtrack_max_ratio = 0.50 [static], [protected]`

Definition at line 460 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::profile_bound_backtrack()`.

**9.23.4.2** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio = 1e-3 [static], [protected]`

Definition at line 461 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.23.4.3** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_ratio = 0.05 [static], [protected]`

Definition at line 459 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.23.4.4** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio = 1.0e-9 [static], [protected]`

Convergence criteria: tolerance for function-value change.

Definition at line 456 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, and `mappel::estimator::IterativeMaximizer< Model >::get_stats()`.

**9.23.4.5** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio = 1.0e-9 [static], [protected]`

Convergence criteria: tolerance of relative step size.

Definition at line 457 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.23.4.6** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::DefaultIterations = 100 [static]`

Definition at line 428 of file estimator.h.

**9.23.4.7** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts [protected], [inherited]`

Definition at line 299 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.23.4.8** `template<class Model> IdxVecT mappel::estimator::IterativeMaximizer< Model>::last_backtrack_idx`  
`[protected]`

Debugging: Stores last set of backtrack\_idx when data.save\_seq==true.

Definition at line 475 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`.

**9.23.4.9** `template<class Model> const int mappel::estimator::IterativeMaximizer< Model>::max_backtracks = 8`  
`[static], [protected]`

Definition at line 462 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::backtrack()`, `mappel::estimator::IterativeMaximizer< Model>::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model>::profile_bound_backtrack()`.

**9.23.4.10** `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::max_iterations`  
`[protected]`

Definition at line 466 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.23.4.11** `template<class Model> int mappel::estimator::ThreadedEstimator< Model>::max_threads`  
`[protected], [inherited]`

Definition at line 324 of file estimator.h.

**9.23.4.12** `template<class Model> const double mappel::estimator::IterativeMaximizer< Model>::min_eigenvalue_correction_delta = 1e-3`  
`[static], [protected]`

Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.

Definition at line 454 of file estimator.h.

**9.23.4.13** `template<class Model> const double mappel::estimator::IterativeMaximizer< Model>::min_profile_bound_residual = 1e-4`  
`[static], [protected]`

Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.

Definition at line 464 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.



**9.23.4.14** `template<class Model > const Model& mappel::estimator::Estimator< Model >::model` [protected],  
[inherited]

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.23.4.15** `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` [protected],  
[inherited]

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.23.4.16** `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected],  
[inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

**9.23.4.17** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads`  
[protected], [inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.

**9.23.4.18** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_backtracks = 0`  
`[protected]`

Definition at line 470 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.23.4.19** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_der_evals = 0`  
`[protected]`

Definition at line 472 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.23.4.20** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_fun_evals = 0`  
`[protected]`

Definition at line 471 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.23.4.21** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_iterations = 0`  
`[protected]`

Definition at line 469 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.23.4.22** `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected]`,  
`[inherited]`

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

The documentation for this class was generated from the following files:

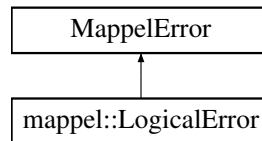
- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.24 mappel::LogicalError Struct Reference

Failure of code or algorithm logic.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/util.h>
```

Inheritance diagram for mappel::LogicalError:



### Public Member Functions

- [LogicalError](#) (std::string message)

#### 9.24.1 Detailed Description

Failure of code or algorithm logic.

Definition at line 104 of file `util.h`.

#### 9.24.2 Constructor & Destructor Documentation

##### 9.24.2.1 mappel::LogicalError::LogicalError ( std::string *message* ) [inline]

Definition at line 106 of file `util.h`.

The documentation for this struct was generated from the following file:

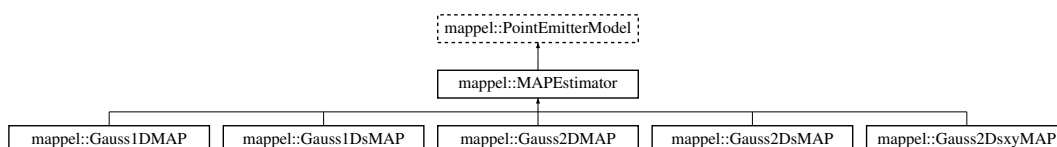
- [util.h](#)

## 9.25 mappel::MAPEstimator Class Reference

A Mixin class to configure a for MLE estimation (null prior).

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/MAPEstimator.h>
```

Inheritance diagram for mappel::MAPEstimator:



## Public Types

- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat

## Public Member Functions

- [StatsT](#) [get\\_stats](#) () const
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT](#) [make\\_param](#) () const
- template<class FillT >  
  [ParamT](#) [make\\_param](#) (FillT fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT](#) [make\\_param\\_mat](#) () const
- template<class FillT >  
  [MatT](#) [make\\_param\\_mat](#) (FillT fill) const
- [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get\\_prior](#) ()
- const CompositeDist & [get\\_prior](#) () const
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- [IdxT](#) [get\\_num\\_hyperparams](#) () const
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- [VecT](#) [get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT](#) [get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT](#) [get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT](#) [sample\\_prior](#) (RngT &rng) const
- [ParamT](#) [sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get\\_lbound](#) () const
- const [ParamT](#) & [get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const

- void `bound_theta` (`ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `bounded_theta` (const `ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `reflected_theta` (const `ParamT` &theta) const
- `BoolVecT` `theta_stack_in_bounds` (const `ParamVecT` &theta) const
- `ParamVecT` `bounded_theta_stack` (const `ParamVecT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamVecT` `reflected_theta_stack` (const `ParamVecT` &theta) const

#### Static Public Member Functions

- static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` size, double pos\_↔ sigma=`default_sigma_pos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos\_↔ beta=`default_beta_pos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=`default_mean_l`↔ l, double kappa=`default_intensity_kappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min\_sigma, double max\_↔ sigma, double alpha=`default_alpha_sigma`)
- static void `set_rng_seed` (`RngSeedT` seed)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

#### Static Public Attributes

- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1
- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2

#### Protected Member Functions

- `MAPEstimator` ()
- `MAPEstimator` (const `MAPEstimator` &o)
- `MAPEstimator` (`MAPEstimator` &&o)
- `MAPEstimator` & `operator=` (const `MAPEstimator` &o)
- `MAPEstimator` & `operator=` (`MAPEstimator` &&o)

#### Protected Attributes

- `CompositeDist` `prior`
- `IdxT` `num_params`
- `IdxT` `num_hyperparams`
- `ParamT` `lbound`
- `ParamT` `ubound`

### 9.25.1 Detailed Description

A Mixin class to configure a for MLE estimation (null prior).

Inheriting from this class modifies the objective function undergoing optimization to use a Null prior, by simply ignoring the effect of the prior on the objective. This which effectively turns the objective function into a pure likelihood function, and the estimator becomes an MLE estimator.

Definition at line 22 of file MAPEstimator.h.

### 9.25.2 Member Typedef Documentation

**9.25.2.1** `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

**9.25.2.2** `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

### 9.25.3 Constructor & Destructor Documentation

**9.25.3.1** `mappel::MAPEstimator::MAPEstimator ( )` [inline],[protected]

Definition at line 24 of file MAPEstimator.h.

**9.25.3.2** `mappel::MAPEstimator::MAPEstimator ( const MAPEstimator & o )` [inline],[protected]

Definition at line 25 of file MAPEstimator.h.

**9.25.3.3** `mappel::MAPEstimator::MAPEstimator ( MAPEstimator && o )` [inline],[protected]

Definition at line 26 of file MAPEstimator.h.

### 9.25.4 Member Function Documentation

**9.25.4.1** `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const`  
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

**9.25.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const** [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

**9.25.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & *theta*, double *epsilon* = bounds\_epsilon ) const** [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

**9.25.4.4 void mappel::PointEmitterModel::check\_param\_shape ( const ParamT & *theta* ) const** [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

**9.25.4.5 void mappel::PointEmitterModel::check\_param\_shape ( const ParamVecT & *theta* ) const** [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

**9.25.4.6 void mappel::PointEmitterModel::check\_psf\_sigma ( double *psf\_sigma* ) const** [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

**9.25.4.7 void mappel::PointEmitterModel::check\_psf\_sigma ( const VecT & *psf\_sigma* ) const** [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

**9.25.4.8** `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.25.4.9** `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` `[inline]`,`[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.25.4.10** `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.25.4.11** `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.25.4.12** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.25.4.13** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline]`,`[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

**9.25.4.14** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline]`,`[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.



**9.25.4.15 StringVecT mappel::PointEmitterModel::get\_param\_names ( ) const** [inline],[inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.25.4.16 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.25.4.17 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.25.4.18 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.25.4.19 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( )** [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.25.4.20 StatsT mappel::PointEmitterModel::get\_stats ( ) const** [inherited]

Definition at line 125 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, mappel::rng\_manager, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DModel::get\_stats(), mappel::Gauss1DsModel::get\_stats(), mappel::Gauss2DModel::get\_stats(), and mappel::Gauss2DsModel::get\_stats().

**9.25.4.21 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const** [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.25.4.22 **bool** mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.25.4.23 **PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline],[inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

9.25.4.24 **template<class FillT > PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( FillT fill ) const [inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.25.4.25 **MatT** mappel::PointEmitterModel::make\_param\_mat ( ) const [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.25.4.26 **template<class FillT > MatT** mappel::PointEmitterModel::make\_param\_mat ( FillT fill ) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.25.4.27 **CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT n ) const [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.25.4.28 **template<class FillT > CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT n, FillT fill ) const [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.25.4.29** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline]`,  
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.25.4.30** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.25.4.31** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_I, double kappa = default_intensity_kappa )` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.25.4.32** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos )` `[static]`, `[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

**9.25.4.33** `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos )` `[static]`, `[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.25.4.34** `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma )` `[static], [inherited]`

Definition at line 104 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.25.4.35** `MAPEstimator& mappel::MAPEstimator::operator= ( const MAPEstimator & o )` `[inline], [protected]`

Definition at line 27 of file `MAPEstimator.h`.

Referenced by `mappel::Gauss1DMP::operator=()`, `mappel::Gauss1DsMP::operator=()`, `mappel::Gauss2DsMP::operator=()`, and `mappel::Gauss2DMP::operator=()`.

**9.25.4.36** `MAPEstimator& mappel::MAPEstimator::operator= ( MAPEstimator && o )` `[inline], [protected]`

Definition at line 28 of file `MAPEstimator.h`.

**9.25.4.37** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const` `[inherited]`

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.25.4.38** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.25.4.39** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )` `[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.25.4.40 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.25.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline],[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.25.4.42 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.25.4.43 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline],[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.25.4.44 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline],[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.25.4.45 `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams )` `[inline],[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

9.25.4.46 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.25.4.47 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.25.4.48 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )` [inherited]

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

9.25.4.49 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ )` [inherited]

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.25.4.50 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` [static],[inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.25.4.51 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

9.25.4.52 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const` [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

9.25.4.53 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const` [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

## 9.25.5 Member Data Documentation

9.25.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static], [inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

9.25.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.25.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.25.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static], [inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.25.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.25.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.25.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.25.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.25.5.9** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.25.5.10** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.



#### 9.25.5.11 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static]`, `[inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

#### 9.25.5.12 `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.25.5.13 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

#### 9.25.5.14 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

#### 9.25.5.15 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

#### 9.25.5.16 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

The documentation for this class was generated from the following file:

- [MAPEstimator.h](#)

### 9.26 mappel::estimator::IterativeMaximizer< Model >::MaximizerData Class Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

#### Public Member Functions

- [MaximizerData](#) (const Model &model, const [ModelDataT](#)< Model > &im, const [StencilT](#)< Model > &s, bool save\_seq=false)
- [MaximizerData](#) (const Model &model, const [ModelDataT](#)< Model > &im, const [StencilT](#)< Model > &s, double rllh, bool save\_seq=false)
- void [record\\_iteration](#) ()
- void [record\\_iteration](#) (const [ParamT](#)< Model > &accepted\_theta)  
*Record an iteration point (derivatives computed)*
- void [record\\_backtrack](#) (double rejected\_rllh)  
*Record a backtracked point (no derivative computations performed) Using the saved theta as the default.*
- void [record\\_backtrack](#) (const [ParamT](#)< Model > &rejected\_theta, double rejected\_rllh)  
*Record a backtracked point (no derivative computations performed)*
- bool [has\\_theta\\_sequence](#) () const  
*Return the saved theta sequence.*
- [IdxT](#) [get\\_sequence\\_len](#) () const
- [ParamVecT](#)< Model > [get\\_theta\\_sequence](#) () const
- [IdxVecT](#) [get\\_backtrack\\_idx](#) () const
- [VecT](#) [get\\_theta\\_sequence\\_rllh](#) () const
- [StencilT](#)< Model > & [stencil](#) ()  
*Get the current stencil.*
- void [set\\_stencil](#) (const [StencilT](#)< Model > &s)
- void [save\\_stencil](#) ()  
*Save the current stencil to the single reserve spot. Overwrites any previously saved stencil. This is used to save a stencil when backtracking.*
- void [restore\\_stencil](#) ()  
*Restore the single reserved stencil to the current stencil spot. Overwrites any previously saved stencil. This is used to restore a last good iterate (and associated stencil data) when backtracking.*
- const [StencilT](#)< Model > & [saved\\_stencil](#) () const

- *Get the saved stencil.*
- const ParamT< Model > & theta () const
- *Get the current stencil's theta.*
- const ParamT< Model > & saved\_theta () const
- *Get the saved stencil's theta.*
- void set\_fixed\_parameters (const IdxVecT &fixed\_parameters\_idx)
- bool has\_fixed\_parameters () const
- IdxT num\_fixed\_parameters () const

#### Public Attributes

- const ModelDataT< Model > & im
- ParamT< Model > grad
- ParamT< Model > step
- double rllh
- int nBacktracks =0
- int nIterations =0
- IdxVecT fixed\_idx
- IdxVecT free\_idx

#### Protected Member Functions

- void expand\_max\_seq\_len ()

#### Protected Attributes

- const IdxT num\_params
- StencilT< Model > s0
- StencilT< Model > s1
- bool current\_stencil
- int max\_seq\_len =0
- int seq\_len =0
- ParamVecT< Model > theta\_seq
- VecT seq\_rllh
- IdxVecT backtrack\_idx

#### Static Protected Attributes

- static const int DefaultMaxSeqLength = 50
- Default maximum length of sequence to prepare to save if debugging.*

#### 9.26.1 Detailed Description

```
template<class Model>
class mappel::estimator::IterativeMaximizer< Model >::MaximizerData
```

Definition at line 477 of file estimator.h.

### 9.26.2 Constructor & Destructor Documentation

**9.26.2.1** `template<class Model > mappel::estimator::IterativeMaximizer< Model >::MaximizerData::MaximizerData ( const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & s, bool save_seq = false )`

Definition at line 738 of file estimator\_impl.h.

**9.26.2.2** `template<class Model > mappel::estimator::IterativeMaximizer< Model >::MaximizerData::MaximizerData ( const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & s, double rllh, bool save_seq = false )`

Definition at line 744 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::DefaultMaxSeqLength`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::max_seq_len`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::methods::objective::rllh()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::seq_rllh`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta_seq`.

### 9.26.3 Member Function Documentation

**9.26.3.1** `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::expand_max_seq_len ( ) [protected]`

Definition at line 766 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::max_seq_len`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::seq_rllh`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta_seq`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

**9.26.3.2** `template<class Model > IdxVecT mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idx ( ) const [inline]`

Definition at line 504 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.26.3.3** `template<class Model > IdxT mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_sequence_len ( ) const [inline]`

Definition at line 502 of file estimator.h.

9.26.3.4 `template<class Model > ParamVecT<Model> mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence ( ) const [inline]`

Definition at line 503 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.26.3.5 `template<class Model > VecT mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh ( ) const [inline]`

Definition at line 505 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.26.3.6 `template<class Model > bool mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters ( ) const [inline]`

Definition at line 528 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.26.3.7 `template<class Model > bool mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_theta_sequence ( ) const [inline]`

Return the saved theta sequence.

Definition at line 501 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.26.3.8 `template<class Model > IdxT mappel::estimator::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters ( ) const [inline]`

Definition at line 529 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.26.3.9 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack ( double rejected_rllh ) [inline]`

Record a backtracked point (no derivative computations performed) Using the saved theta as the default.

Definition at line 496 of file estimator.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`.

**9.26.3.10** `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack ( const ParamT< Model > & rejected_theta, double rejected_rllh )`

Record a backtracked point (no derivative computations performed)

Definition at line 787 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::expand_max_seq_len()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::max_seq_len`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nBacktracks`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::seq_len`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::seq_rllh`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta_seq`.

**9.26.3.11** `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration ( ) [inline]`

Definition at line 492 of file estimator.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::MaximizerData()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

**9.26.3.12** `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration ( const ParamT< Model > & accepted_theta )`

Record an iteration point (derivatives computed)

Definition at line 775 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::expand_max_seq_len()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::max_seq_len`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nIterations`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::seq_len`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::seq_rllh`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta_seq`.

**9.26.3.13** `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil ( ) [inline]`

Restore the single reserved stencil to the current stencil spot. Overwrites any previously saved stencil. This is used to restore a last good iterate (and associated stencil data) when backtracking.

Definition at line 519 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.26.3.14 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil ( )`  
`[inline]`

Save the current stencil to the single reserve spot. Overwrites any previously saved stencil. This is used to save a stencil when backtracking.

Definition at line 514 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.26.3.15 `template<class Model > const StencilT<Model>& mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_stencil ( ) const` `[inline]`

Get the saved stencil.

Definition at line 521 of file estimator.h.

9.26.3.16 `template<class Model > const ParamT<Model>& mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta ( ) const` `[inline]`

Get the saved stencil's theta.

Definition at line 525 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.26.3.17 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters ( const IdxVecT & fixed_parameters_idxxs )`

Definition at line 800 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::fixed_idxxs`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::free_idxxs`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::num_params`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.26.3.18 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil ( const StencilT< Model > & s )` `[inline]`

Definition at line 509 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.26.3.19** `template<class Model> StencilT<Model>& mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil ( ) [inline]`

Get the current stencil.

Definition at line 508 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::backtrack()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model>::profile_bound_backtrack()`.

**9.26.3.20** `template<class Model> const ParamT<Model>& mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta ( ) const [inline]`

Get the current stencil's theta.

Definition at line 523 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::backtrack()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model>::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model>::profile_bound_backtrack()`.

## 9.26.4 Member Data Documentation

**9.26.4.1** `template<class Model> IdxVecT mappel::estimator::IterativeMaximizer< Model>::MaximizerData::backtrack_idx [protected]`

Definition at line 540 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::expand_max_seq_len()`, `mappel::estimator::IterativeMaximizer< Model>::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::MaximizerData()`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_backtrack()`.

**9.26.4.2** `template<class Model> bool mappel::estimator::IterativeMaximizer< Model>::MaximizerData::current_stencil [protected]`

Definition at line 534 of file estimator.h.

**9.26.4.3** `template<class Model> const int mappel::estimator::IterativeMaximizer< Model>::MaximizerData::DefaultMaxSeqLength = 50 [static], [protected]`

Default maximum length of sequence to prepare to save if debugging.

Definition at line 531 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::MaximizerData()`.



#### 9.26.4.4 `template<class Model > IdxVecT mappel::estimator::IterativeMaximizer< Model >::MaximizerData::fixed_idx`

Definition at line 487 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`.

#### 9.26.4.5 `template<class Model > IdxVecT mappel::estimator::IterativeMaximizer< Model >::MaximizerData::free_idx`

Definition at line 487 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`.

#### 9.26.4.6 `template<class Model > ParamT<Model> mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`

Definition at line 480 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

#### 9.26.4.7 `template<class Model > const ModelDataT<Model>& mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`

Definition at line 479 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

#### 9.26.4.8 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::MaximizerData::max_seq_len = 0` `[protected]`

Definition at line 536 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::expand_max_seq_len()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::MaximizerData()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

#### 9.26.4.9 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nBacktracks = 0`

Definition at line 484 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.26.4.10** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nIterations =0`

Definition at line 485 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.26.4.11** `template<class Model > const IdxT mappel::estimator::IterativeMaximizer< Model >::MaximizerData::num_params [protected]`

Definition at line 532 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`.

**9.26.4.12** `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`

Definition at line 482 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::MaximizerData()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

**9.26.4.13** `template<class Model > StencilT<Model> mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s0 [protected]`

Definition at line 533 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.26.4.14** `template<class Model > StencilT<Model> mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s1 [protected]`

Definition at line 533 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.26.4.15** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::MaximizerData::seq_len =0 [protected]`

Definition at line 537 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

9.26.4.16 `template<class Model> VecT mappel::estimator::IterativeMaximizer< Model>::MaximizerData::seq_rllh`  
`[protected]`

Definition at line 539 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::expand_max_seq_len()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::MaximizerData()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_iteration()`.

9.26.4.17 `template<class Model> ParamT<Model> mappel::estimator::IterativeMaximizer< Model>::MaximizerData::step`

Definition at line 481 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::backtrack()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model>::profile_bound_backtrack()`.

9.26.4.18 `template<class Model> ParamVecT<Model> mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta_seq` `[protected]`

Definition at line 538 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::expand_max_seq_len()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::MaximizerData()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_backtrack()`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_iteration()`.

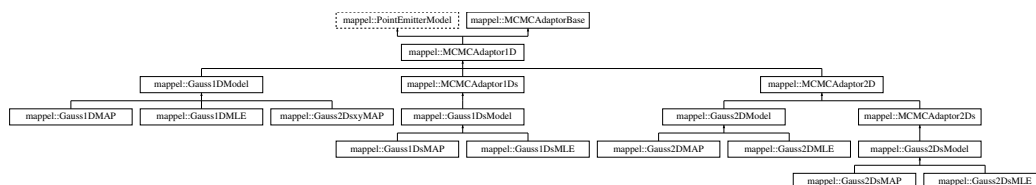
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.27 mappel::MCMCAdaptor1D Class Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/MCMCAdaptor1D.h>
```

Inheritance diagram for `mappel::MCMCAdaptor1D`:



## Public Types

- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat

## Public Member Functions

- void [sample\\_mcmc\\_candidate](#) (IdxT sample\_index, [ParamT](#) &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) (IdxT sample\_index, [ParamT](#) &candidate, const [IdxVecT](#) &fixed\_parameters\_↵ mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)
- void [set\\_background\\_mcmc\\_sampling](#) (double eta\_bg=-1)
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT](#) [make\\_param](#) () const
- template<class FillT >  
  [ParamT](#) [make\\_param](#) (FillT fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) (IdxT n) const
- template<class FillT >  
  [ParamVecT](#) [make\\_param\\_stack](#) (IdxT n, FillT fill) const
- [MatT](#) [make\\_param\\_mat](#) () const
- template<class FillT >  
  [MatT](#) [make\\_param\\_mat](#) (FillT fill) const
- [CubeT](#) [make\\_param\\_mat\\_stack](#) (IdxT n) const
- template<class FillT >  
  [CubeT](#) [make\\_param\\_mat\\_stack](#) (IdxT n, FillT fill) const
- CompositeDist & [get\\_prior](#) ()
- const CompositeDist & [get\\_prior](#) () const
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- [IdxT](#) [get\\_num\\_hyperparams](#) () const
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- [VecT](#) [get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT](#) [get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT](#) [get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT](#) [sample\\_prior](#) (RngT &rng) const
- [ParamT](#) [sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)

- void `set_ubound` (const `ParamT` &`ubound`)
- const `ParamT` & `get_lbound` () const
- const `ParamT` & `get_ubound` () const
- bool `theta_in_bounds` (const `ParamT` &`theta`) const
- void `bound_theta` (`ParamT` &`theta`, double `epsilon=bounds_epsilon`) const
- `ParamT` `bounded_theta` (const `ParamT` &`theta`, double `epsilon=bounds_epsilon`) const
- `ParamT` `reflected_theta` (const `ParamT` &`theta`) const
- `BoolVecT` `theta_stack_in_bounds` (const `ParamVecT` &`theta`) const
- `ParamVecT` `bounded_theta_stack` (const `ParamVecT` &`theta`, double `epsilon=bounds_epsilon`) const
- `ParamVecT` `reflected_theta_stack` (const `ParamVecT` &`theta`) const
- void `set_mcmc_sigma_scale` (double `scale`)
- double `get_mcmc_sigma_scale` () const
- `IdxT` `get_mcmc_num_phases` () const

#### Static Public Member Functions

- static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` `size`, double `pos_↵` `sigma=default_sigma_pos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` `size`, double `pos_↵` `beta=default_beta_pos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double `mean=default_mean_↵` `l`, double `kappa=default_intensity_kappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double `min_sigma`, double `max_↵` `sigma`, double `alpha=default_alpha_sigma`)
- static void `set_rng_seed` (`RngSeedT` `seed`)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

#### Static Public Attributes

- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1
- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2
- static const double `global_default_mcmc_sigma_scale` = 0.05
- static const double `global_max_mcmc_sigma_scale` = 0.5

### Protected Member Functions

- [MCMCAdaptor1D](#) ()
- [MCMCAdaptor1D](#) (double [sigma\\_scale](#))
- [MCMCAdaptor1D](#) (const [MCMCAdaptor1D](#) &o)
- [MCMCAdaptor1D](#) ([MCMCAdaptor1D](#) &&o)
- [MCMCAdaptor1D](#) & [operator=](#) (const [MCMCAdaptor1D](#) &o)
- [MCMCAdaptor1D](#) & [operator=](#) ([MCMCAdaptor1D](#) &&o)
- [StatsT](#) [get\\_stats](#) () const
- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) [num\\_phases](#))

### Protected Attributes

- double [eta\\_x](#) =0
- double [eta\\_l](#) =0
- double [eta\\_bg](#) =0
- CompositeDist [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [IdxT](#) [num\\_phases](#)
- double [sigma\\_scale](#)

#### 9.27.1 Detailed Description

Definition at line 15 of file [MCMCAdaptor1D.h](#).

#### 9.27.2 Member Typedef Documentation

9.27.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file [PointEmitterModel.h](#).

9.27.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file [PointEmitterModel.h](#).

#### 9.27.3 Constructor & Destructor Documentation

9.27.3.1 `mappel::MCMCAdaptor1D::MCMCAdaptor1D ( )` [protected]

Definition at line 11 of file [MCMCAdaptor1D.cpp](#).

### 9.27.3.2 mappel::MCMCAdaptor1D::MCMCAdaptor1D ( double *sigma\_scale* ) [explicit], [protected]

Definition at line 14 of file MCMCAdaptor1D.cpp.

References `eta_x`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, `set_↔background_mcmc_sampling()`, `set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

### 9.27.3.3 mappel::MCMCAdaptor1D::MCMCAdaptor1D ( const MCMCAdaptor1D & o ) [protected]

Definition at line 24 of file MCMCAdaptor1D.cpp.

References `eta_bg`, `eta_l`, and `eta_x`.

### 9.27.3.4 mappel::MCMCAdaptor1D::MCMCAdaptor1D ( MCMCAdaptor1D && o ) [protected]

Definition at line 33 of file MCMCAdaptor1D.cpp.

References `eta_bg`, `eta_l`, and `eta_x`.

## 9.27.4 Member Function Documentation

### 9.27.4.1 void mappel::PointEmitterModel::bound\_theta ( ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::Point↔EmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

### 9.27.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::Point↔EmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

### 9.27.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

9.27.4.4 `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.27.4.5 `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

9.27.4.6 `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

9.27.4.7 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

9.27.4.8 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline], [inherited]

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` [inline], [inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.



9.27.4.10 `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by set\_background\_mcmc\_sampling(), and set\_intensity\_mcmc\_sampling().

9.27.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and set\_background\_mcmc\_sampling().

9.27.4.13 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

9.27.4.14 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

9.27.4.15 `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline]`, `[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

9.27.4.16 `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline]`, `[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

#### 9.27.4.17 `StringVecT mappel::PointEmitterModel::get_param_names ( ) const [inline],[inherited]`

Definition at line 248 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

#### 9.27.4.18 `CompositeDist & mappel::PointEmitterModel::get_prior ( ) [inline],[inherited]`

Definition at line 200 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

#### 9.27.4.19 `const CompositeDist & mappel::PointEmitterModel::get_prior ( ) const [inline],[inherited]`

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

#### 9.27.4.20 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ( ) [static],[inherited]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

#### 9.27.4.21 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ( ) [static],[inherited]`

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

#### 9.27.4.22 `StatsT mappel::MCMCAdaptor1D::get_stats ( ) const [protected]`

Definition at line 98 of file `MCMCAdaptor1D.cpp`.

References `eta_bg`, `eta_l`, `eta_x`, and `mappel::MCMCAdaptorBase::get_stats()`.

Referenced by `mappel::MCMCAdaptor1Ds::get_stats()`, `mappel::MCMCAdaptor2D::get_stats()`, and `mappel::Gauss1DModel::get_stats()`.

#### 9.27.4.23 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound ( ) const [inline],[inherited]`

Definition at line 216 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DsxyModel::get_max_sigma_ratio()`, `mappel::Gauss2DsModel::get_max_sigma_ratio()`, `MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `set_background_mcmc_sampling()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

9.27.4.24 `bool mappel::PointEmitterModel::has_hyperparam ( const std::string & name ) const` `[inline],[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( ) const` `[inline],[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

9.27.4.26 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const` `[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.27.4.27 `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.27.4.28 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.27.4.29 `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.27.4.30 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.27.4.31** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline]`,  
`[inherited]`

Definition at line 168 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.27.4.32** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.27.4.33** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_I, double kappa = default_intensity_kappa )` `[static]`, `[inherited]`

Definition at line 98 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.27.4.34** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos )` `[static]`, `[inherited]`

Definition at line 92 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

**9.27.4.35** `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos )` `[static]`, `[inherited]`

Definition at line 85 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.27.4.36** `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma )` `[static], [inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.27.4.37** `MCMCAdaptor1D & mappel::MCMCAdaptor1D::operator= ( const MCMCAdaptor1D & o )` `[protected]`

Definition at line 42 of file MCMCAdaptor1D.cpp.

References `eta_bg`, `eta_l`, and `eta_x`.

Referenced by `mappel::MCMCAdaptor1Ds::operator=()`, `mappel::MCMCAdaptor2D::operator=()`, and `mappel::Gauss1DModel::operator=()`.

**9.27.4.38** `MCMCAdaptor1D & mappel::MCMCAdaptor1D::operator= ( MCMCAdaptor1D && o )` `[protected]`

Definition at line 53 of file MCMCAdaptor1D.cpp.

References `eta_bg`, `eta_l`, and `eta_x`.

**9.27.4.39** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const` `[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.27.4.40** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.27.4.41** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )` `[inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.27.4.42 `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const`

Definition at line 108 of file MCMCAdaptor1D.cpp.

References `eta_bg`, `eta_l`, `eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.27.4.43 `void mappel::MCMCAdaptor1D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const`

Definition at line 122 of file MCMCAdaptor1D.cpp.

References `eta_bg`, `eta_l`, `eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

9.27.4.44 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const [inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.27.4.45 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const [inline],[inherited]`

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.27.4.46 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `MCMCAdaptor1D()`.

9.27.4.47 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound, const ParamT & ubound ) [inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.27.4.48 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc ) [inline], [inherited]`

Definition at line 260 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.49 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.50 `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams ) [inline], [inherited]`

Definition at line 220 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::set\_hyperparams(), and mappel::Gauss2DsModel::set\_hyperparams().

9.27.4.51 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 )`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_mean\_I, eta\_I, mappel::PointEmitterModel::get\_hyperparam\_value(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by MCMCAdaptor1D().

9.27.4.52 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.27.4.53 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases ) [protected], [inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

Referenced by mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), and mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds().

9.27.4.54 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` `[inherited]`

Definition at line 39 of file `MCMCAdaptorBase.cpp`.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_↵_scale`.

9.27.4.55 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` `[inline],[inherited]`

Definition at line 252 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.27.4.56 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )` `[inherited]`

Definition at line 158 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::Point_↵EmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

9.27.4.57 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ )` `[inherited]`

Definition at line 149 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::Point_↵EmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.27.4.58 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` `[static],[inherited]`

Definition at line 110 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

9.27.4.59 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` `[inherited]`

Definition at line 237 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitter_↵Model::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.



9.27.4.60 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const` [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

9.27.4.61 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const` [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

## 9.27.5 Member Data Documentation

9.27.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static], [inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

9.27.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.27.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.27.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static], [inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.27.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.27.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `set_intensity_mcmc_sampling()`.

**9.27.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `set_background_mcmc_sampling()`.

**9.27.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.27.5.9** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.27.5.10** `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `get_stats()`, `MCMCAdaptor1D()`, `operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `sample_mcmc_candidate()`, and `set_background_mcmc_sampling()`.

**9.27.5.11 double mappel::MCMCAdaptor1D::eta\_l=0 [protected]**

The standard deviation for the normally distributed perturbation to theta\_l in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by get\_stats(), MCMCAdaptor1D(), operator=(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), sample\_mcmc\_candidate(), and set\_intensity\_mcmc\_sampling().

**9.27.5.12 double mappel::MCMCAdaptor1D::eta\_x=0 [protected]**

The standard deviation for the normally distributed perturbation to theta\_x in the random walk MCMC sampling

Definition at line 31 of file MCMCAdaptor1D.h.

Referenced by get\_stats(), MCMCAdaptor1D(), operator=(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), and sample\_mcmc\_candidate().

**9.27.5.13 const double mappel::MCMCAdaptorBase::global\_default\_mcmc\_sigma\_scale = 0.05 [static],[inherited]**

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds().

**9.27.5.14 const double mappel::MCMCAdaptorBase::global\_max\_mcmc\_sigma\_scale = 0.5 [static],[inherited]**

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::MCMCAdaptorBase(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

**9.27.5.15 const double mappel::PointEmitterModel::global\_max\_psf\_sigma = 1E2 [static],[inherited]**

Global maximum for any psf\_sigma. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::check\_psf\_sigma().

**9.27.5.16 const double mappel::PointEmitterModel::global\_min\_psf\_sigma = 1E-1 [static],[inherited]**

Global minimum for any psf\_sigma. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::check\_psf\_sigma().

#### 9.27.5.17 ParamT mappel::PointEmitterModel::lbound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.27.5.18 IdxT mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.27.5.19 IdxT mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.27.5.20 IdxT mappel::MCMCAdaptorBase::num\_phases [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().

#### 9.27.5.21 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

9.27.5.22 `double mappel::MCMCAdaptorBase::sigma_scale` [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `set_background_mcmc_sampling()`, `set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

9.27.5.23 `ParamT mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

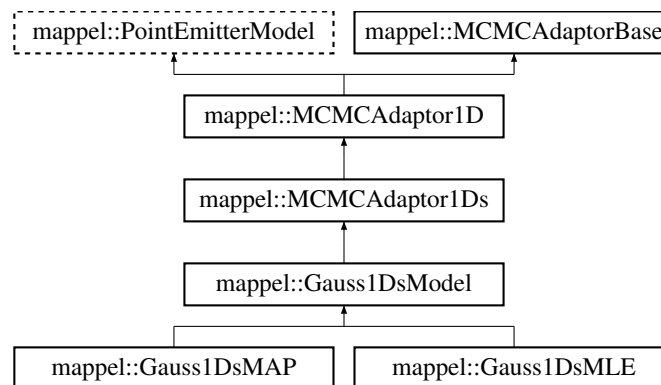
The documentation for this class was generated from the following files:

- [MCMCAdaptor1D.h](#)
- [MCMCAdaptor1D.cpp](#)

## 9.28 mappel::MCMCAdaptor1Ds Class Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/MCMCAdaptor1Ds.h>
```

Inheritance diagram for `mappel::MCMCAdaptor1Ds`:



### Public Types

- using `ParamT` = `arma::vec`
- using `ParamVecT` = `arma::mat`

## Public Member Functions

- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, const [IdxVecT](#) &fixed\_parameters\_↔ mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)
- void [set\\_background\\_mcmc\\_sampling](#) (double eta\_bg=-1)
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT](#) [make\\_param](#) () const
- template<class FillT >  
  [ParamT](#) [make\\_param](#) (FillT fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT](#) [make\\_param\\_mat](#) () const
- template<class FillT >  
  [MatT](#) [make\\_param\\_mat](#) (FillT fill) const
- [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get\\_prior](#) ()
- const CompositeDist & [get\\_prior](#) () const
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- [IdxT](#) [get\\_num\\_hyperparams](#) () const
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- [VecT](#) [get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT](#) [get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT](#) [get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT](#) [sample\\_prior](#) (RngT &rng) const
- [ParamT](#) [sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get\\_lbound](#) () const
- const [ParamT](#) & [get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT](#) [bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const

- [ParamT](#) [reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT](#) [theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT](#) [bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT](#) [reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- void [set\\_mcmc\\_sigma\\_scale](#) (double scale)
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- [IdxT](#) [get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Member Functions

- static [prior\\_hessian::TruncatedNormalDist](#) [make\\_prior\\_component\\_position\\_normal](#) ([IdxT](#) size, double pos\_↵, double sigma=[default\\_sigma\\_pos](#))
- static [prior\\_hessian::ScaledSymmetricBetaDist](#) [make\\_prior\\_component\\_position\\_beta](#) ([IdxT](#) size, double pos\_↵, double beta=[default\\_beta\\_pos](#))
- static [prior\\_hessian::TruncatedGammaDist](#) [make\\_prior\\_component\\_intensity](#) (double mean=[default\\_mean\\_l](#)↵, double kappa=[default\\_intensity\\_kappa](#))
- static [prior\\_hessian::TruncatedParetoDist](#) [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵sigma, double alpha=[default\\_alpha\\_sigma](#))
- static void [set\\_rng\\_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()

#### Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5

#### Protected Member Functions

- [MCMCAdaptor1Ds](#) ()
- [MCMCAdaptor1Ds](#) (double [sigma\\_scale](#))
- [MCMCAdaptor1Ds](#) (const [MCMCAdaptor1Ds](#) &o)
- [MCMCAdaptor1Ds](#) ([MCMCAdaptor1Ds](#) &&o)
- [MCMCAdaptor1Ds](#) & [operator=](#) (const [MCMCAdaptor1Ds](#) &o)
- [MCMCAdaptor1Ds](#) & [operator=](#) ([MCMCAdaptor1Ds](#) &&o)
- [StatsT](#) [get\\_stats](#) () const
- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) num\_phases)

## Protected Attributes

- double `eta_sigma` == -1
- double `eta_x` = 0
- double `eta_l` = 0
- double `eta_bg` = 0
- CompositeDist `prior`
- `IdxT` `num_params`
- `IdxT` `num_hyperparams`
- `ParamT` `lbound`
- `ParamT` `ubound`
- `IdxT` `num_phases`
- double `sigma_scale`

### 9.28.1 Detailed Description

Definition at line 14 of file `MCMCAdaptor1Ds.h`.

### 9.28.2 Member Typedef Documentation

#### 9.28.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file `PointEmitterModel.h`.

#### 9.28.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file `PointEmitterModel.h`.

### 9.28.3 Constructor & Destructor Documentation

#### 9.28.3.1 `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds ( )` [protected]

Definition at line 11 of file `MCMCAdaptor1Ds.cpp`.

References `mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale`.

#### 9.28.3.2 `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds ( double sigma_scale )` [explicit],[protected]

Definition at line 14 of file `MCMCAdaptor1Ds.cpp`.

References `eta_sigma`, `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`, and `mappel::MCMCAdaptorBase::sigma_scale`.



### 9.28.3.3 mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds ( const MCMCAdaptor1Ds & o ) [protected]

Definition at line 21 of file MCMCAdaptor1Ds.cpp.

References eta\_sigma.

### 9.28.3.4 mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds ( MCMCAdaptor1Ds && o ) [protected]

Definition at line 28 of file MCMCAdaptor1Ds.cpp.

References eta\_sigma.

## 9.28.4 Member Function Documentation

### 9.28.4.1 void mappel::PointEmitterModel::bound\_theta ( ParamT & theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

### 9.28.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT & theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

### 9.28.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

### 9.28.4.4 void mappel::PointEmitterModel::check\_param\_shape ( const ParamT & theta ) const [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.28.4.5 `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

9.28.4.6 `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

9.28.4.7 `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

9.28.4.8 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline], [inherited]

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.28.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` [inline], [inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.28.4.10 `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` [inline], [inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.28.4.11** `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.28.4.12** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.28.4.13** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file `MCMCAdaptorBase.cpp`.

References `mappel::MCMCAdaptorBase::num_phases`.

**9.28.4.14** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file `MCMCAdaptorBase.cpp`.

References `mappel::MCMCAdaptorBase::sigma_scale`.

**9.28.4.15** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline]`, `[inherited]`

Definition at line 208 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_hyperparams`.

**9.28.4.16** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline]`, `[inherited]`

Definition at line 160 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.28.4.17** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const` `[inline]`, `[inherited]`

Definition at line 248 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

#### 9.28.4.18 CompositeDist & mappel::PointEmitterModel::get\_prior ( ) [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

#### 9.28.4.19 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

#### 9.28.4.20 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( ) [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

#### 9.28.4.21 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( ) [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

#### 9.28.4.22 StatsT mappel::MCMCAdaptor1Ds::get\_stats ( ) const [protected]

Definition at line 51 of file MCMCAdaptor1Ds.cpp.

References eta\_sigma, and mappel::MCMCAdaptor1D::get\_stats().

Referenced by mappel::Gauss1DsModel::get\_stats().

#### 9.28.4.23 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

#### 9.28.4.24 bool mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.28.4.25** `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( ) const` `[inline],[inherited]`

Definition at line 164 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

**9.28.4.26** `template<class FillT> PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const` `[inherited]`

Definition at line 181 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.28.4.27** `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.28.4.28** `template<class FillT> MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.28.4.29** `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.28.4.30** `template<class FillT> CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 196 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.28.4.31** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.28.4.32** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const [inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.28.4.33** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_I, double kappa = default_intensity_kappa ) [static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_I.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

**9.28.4.34** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos ) [static],[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), and mappel::Gauss2DsModel::make\_prior\_beta\_position().

**9.28.4.35** `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos ) [static],[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

**9.28.4.36** `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma ) [static],[inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

**9.28.4.37 MCMCAdaptor1Ds & mappel::MCMCAdaptor1Ds::operator= ( const MCMCAdaptor1Ds & o )** [protected]

Definition at line 35 of file MCMCAdaptor1Ds.cpp.

References `eta_sigma`, and `mappel::MCMCAdaptor1D::operator=()`.

Referenced by `mappel::Gauss1DsModel::operator=()`.

**9.28.4.38 MCMCAdaptor1Ds & mappel::MCMCAdaptor1Ds::operator= ( MCMCAdaptor1Ds && o )** [protected]

Definition at line 43 of file MCMCAdaptor1Ds.cpp.

References `eta_sigma`, and `mappel::MCMCAdaptor1D::operator=()`.

**9.28.4.39 PointEmitterModel::ParamT mappel::PointEmitterModel::reflected\_theta ( const ParamT & theta ) const**  
[inherited]

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.28.4.40 PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected\_theta\_stack ( const ParamVecT & theta ) const** [inherited]

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.28.4.41 void mappel::PointEmitterModel::rename\_hyperparam ( const std::string & old\_name, const std::string & new\_name )**  
[inline], [inherited]

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.28.4.42 void mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate ( IdxT sample\_index, ParamT & candidate, double step\_scale = 1.0 ) const**

Definition at line 59 of file MCMCAdaptor1Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.28.4.43** `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const`

Definition at line 77 of file MCMCAdaptor1Ds.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.28.4.44** `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const` `[inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.28.4.45** `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline]`, `[inherited]`

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

**9.28.4.46** `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` `[inherited]`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

**9.28.4.47** `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )` `[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.28.4.48** `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline]`, `[inherited]`

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.



9.28.4.49 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.28.4.50 `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams ) [inline], [inherited]`

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

9.28.4.51 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_I = -1 ) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.28.4.52 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.28.4.53 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases ) [protected], [inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

9.28.4.54 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale ) [inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

**9.28.4.55** `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc ) [inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.28.4.56** `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ ) [inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

**9.28.4.57** `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ ) [inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.28.4.58** `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

**9.28.4.59** `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

**9.28.4.60** `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxyModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

9.28.4.61 **BoolVecT** mappel::PointEmitterModel::theta\_stack\_in\_bounds ( const ParamVecT & *theta* ) const [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::theta\_in\_bounds().

## 9.28.5 Member Data Documentation

9.28.5.1 **const double** mappel::PointEmitterModel::bounds\_epsilon = 1.0E-6 [static], [inherited]

Distance from the boundary to constrain in bound\_theta and bounded\_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss2↔DsModel::set\_max\_sigma\_ratio(), and mappel::PointEmitterModel::set\_ubound().

9.28.5.2 **const double** mappel::PointEmitterModel::default\_alpha\_sigma = 2 [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.28.5.3 **const double** mappel::PointEmitterModel::default\_beta\_pos = 3 [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.28.5.4 **const double** mappel::PointEmitterModel::default\_intensity\_kappa = 2 [static], [inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

9.28.5.5 **const double** mappel::PointEmitterModel::default\_max\_l = INFINITY [static], [inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::make\_prior\_component\_intensity().

9.28.5.6 **const double** mappel::PointEmitterModel::default\_mean\_l = 300 [static], [inherited]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.28.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.28.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.28.5.9** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.28.5.10** `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.28.5.11** `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.28.5.12 double mappel::MCMCAdaptor1Ds::eta\_sigma = -1** [protected]

The standard deviation for the normally distributed perturbation to theta\_bg in the random walk MCMC sampling

Definition at line 28 of file MCMCAdaptor1Ds.h.

Referenced by get\_stats(), MCMCAdaptor1Ds(), operator=(), and sample\_mcmc\_candidate().

**9.28.5.13 double mappel::MCMCAdaptor1D::eta\_x = 0** [protected], [inherited]

The standard deviation for the normally distributed perturbation to theta\_x in the random walk MCMC sampling

Definition at line 31 of file MCMCAdaptor1D.h.

Referenced by mappel::MCMCAdaptor1D::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1D::operator=(), sample\_mcmc\_candidate(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), and mappel::MCMCAdaptor1D::sample\_mcmc\_candidate().

**9.28.5.14 const double mappel::MCMCAdaptorBase::global\_default\_mcmc\_sigma\_scale = 0.05** [static], [inherited]

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds().

**9.28.5.15 const double mappel::MCMCAdaptorBase::global\_max\_mcmc\_sigma\_scale = 0.5** [static], [inherited]

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::MCMCAdaptorBase(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

**9.28.5.16 const double mappel::PointEmitterModel::global\_max\_psf\_sigma = 1E2** [static], [inherited]

Global maximum for any psf\_sigma. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::check\_psf\_sigma().

**9.28.5.17 const double mappel::PointEmitterModel::global\_min\_psf\_sigma = 1E-1** [static], [inherited]

Global minimum for any psf\_sigma. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::check\_psf\_sigma().

#### 9.28.5.18 ParamT mappel::PointEmitterModel::lbound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.28.5.19 IdxT mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.28.5.20 IdxT mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.28.5.21 IdxT mappel::MCMCAdaptorBase::num\_phases [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().

#### 9.28.5.22 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

#### 9.28.5.23 double mappel::MCMCAdaptorBase::sigma\_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

#### 9.28.5.24 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

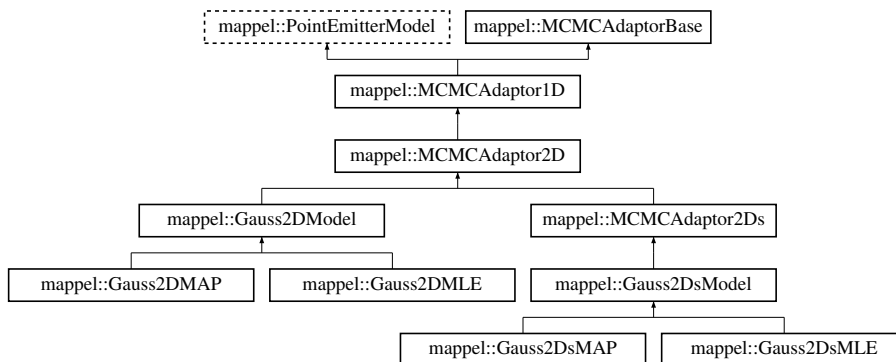
The documentation for this class was generated from the following files:

- [MCMCAdaptor1Ds.h](#)
- [MCMCAdaptor1Ds.cpp](#)

## 9.29 mappel::MCMCAdaptor2D Class Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/MCMCAdaptor2D.h>
```

Inheritance diagram for mappel::MCMCAdaptor2D:



### Public Types

- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat

## Public Member Functions

- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, double step\_scale=1.0) const
- void [sample\\_mcmc\\_candidate](#) ([IdxT](#) sample\_index, [ParamT](#) &candidate, const [IdxVecT](#) &fixed\_parameters\_↔ mask, double step\_scale=1.0) const
- void [set\\_intensity\\_mcmc\\_sampling](#) (double eta\_l=-1)
- void [set\\_background\\_mcmc\\_sampling](#) (double eta\_bg=-1)
- [IdxT](#) [get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT](#) [make\\_param](#) () const
- template<class FillT >  
  [ParamT](#) [make\\_param](#) (FillT fill) const
- [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [ParamVecT](#) [make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT](#) [make\\_param\\_mat](#) () const
- template<class FillT >  
  [MatT](#) [make\\_param\\_mat](#) (FillT fill) const
- [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [CubeT](#) [make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get\\_prior](#) ()
- const CompositeDist & [get\\_prior](#) () const
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- [IdxT](#) [get\\_num\\_hyperparams](#) () const
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- [VecT](#) [get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT](#) [get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT](#) [get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT](#) [sample\\_prior](#) (RngT &rng) const
- [ParamT](#) [sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get\\_lbound](#) () const
- const [ParamT](#) & [get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT](#) [bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const



- [ParamT](#) [reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT](#) [theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT](#) [bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT](#) [reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const
- void [set\\_mcmc\\_sigma\\_scale](#) (double scale)
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- [IdxT](#) [get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Member Functions

- static [prior\\_hessian::TruncatedNormalDist](#) [make\\_prior\\_component\\_position\\_normal](#) ([IdxT](#) size, double pos\_↵ sigma=[default\\_sigma\\_pos](#))
- static [prior\\_hessian::ScaledSymmetricBetaDist](#) [make\\_prior\\_component\\_position\\_beta](#) ([IdxT](#) size, double pos\_↵ beta=[default\\_beta\\_pos](#))
- static [prior\\_hessian::TruncatedGammaDist](#) [make\\_prior\\_component\\_intensity](#) (double mean=[default\\_mean\\_↵ I](#), double kappa=[default\\_intensity\\_kappa](#))
- static [prior\\_hessian::TruncatedParetoDist](#) [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵ sigma, double alpha=[default\\_alpha\\_sigma](#))
- static void [set\\_rng\\_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()

#### Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_I](#) = 300
- static const double [default\\_max\\_I](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2
- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5

#### Protected Member Functions

- [MCMCAdaptor2D](#) ()
- [MCMCAdaptor2D](#) (double [sigma\\_scale](#))
- [MCMCAdaptor2D](#) (const [MCMCAdaptor2D](#) &o)
- [MCMCAdaptor2D](#) ([MCMCAdaptor2D](#) &&o)
- [MCMCAdaptor2D](#) & [operator=](#) (const [MCMCAdaptor2D](#) &o)
- [MCMCAdaptor2D](#) & [operator=](#) ([MCMCAdaptor2D](#) &&o)
- [StatsT](#) [get\\_stats](#) () const
- void [set\\_mcmc\\_num\\_phases](#) ([IdxT](#) num\_phases)

## Protected Attributes

- double `eta_y` =0
- double `eta_x` =0
- double `eta_l` =0
- double `eta_bg` =0
- CompositeDist `prior`
- `IdxT` `num_params`
- `IdxT` `num_hyperparams`
- `ParamT` `lbound`
- `ParamT` `ubound`
- `IdxT` `num_phases`
- double `sigma_scale`

### 9.29.1 Detailed Description

Definition at line 14 of file `MCMCAdaptor2D.h`.

### 9.29.2 Member Typedef Documentation

#### 9.29.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file `PointEmitterModel.h`.

#### 9.29.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file `PointEmitterModel.h`.

### 9.29.3 Constructor & Destructor Documentation

#### 9.29.3.1 `mappel::MCMCAdaptor2D::MCMCAdaptor2D ( )` `[protected]`

Definition at line 11 of file `MCMCAdaptor2D.cpp`.

References `mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale`.

#### 9.29.3.2 `mappel::MCMCAdaptor2D::MCMCAdaptor2D ( double sigma_scale )` `[explicit], [protected]`

Definition at line 14 of file `MCMCAdaptor2D.cpp`.

References `eta_y`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

### 9.29.3.3 mappel::MCMCAdaptor2D::MCMCAdaptor2D ( const MCMCAdaptor2D & o ) [protected]

Definition at line 22 of file MCMCAdaptor2D.cpp.

References [eta\\_y](#).

### 9.29.3.4 mappel::MCMCAdaptor2D::MCMCAdaptor2D ( MCMCAdaptor2D && o ) [protected]

Definition at line 27 of file MCMCAdaptor2D.cpp.

References [eta\\_y](#).

## 9.29.4 Member Function Documentation

### 9.29.4.1 void mappel::PointEmitterModel::bound\_theta ( ParamT & theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References [mappel::PointEmitterModel::check\\_param\\_shape\(\)](#), [mappel::PointEmitterModel::lbound](#), [mappel::PointEmitterModel::num\\_params](#), and [mappel::PointEmitterModel::ubound](#).

### 9.29.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT & theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References [mappel::PointEmitterModel::check\\_param\\_shape\(\)](#), [mappel::PointEmitterModel::lbound](#), [mappel::PointEmitterModel::num\\_params](#), and [mappel::PointEmitterModel::ubound](#).

Referenced by [mappel::PointEmitterModel::bounded\\_theta\\_stack\(\)](#).

### 9.29.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & theta, double epsilon = bounds\_epsilon ) const [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References [mappel::PointEmitterModel::bounded\\_theta\(\)](#), [mappel::PointEmitterModel::check\\_param\\_shape\(\)](#), and [mappel::PointEmitterModel::make\\_param\\_stack\(\)](#).

### 9.29.4.4 void mappel::PointEmitterModel::check\_param\_shape ( const ParamT & theta ) const [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References [mappel::PointEmitterModel::num\\_params](#).

Referenced by [mappel::PointEmitterModel::bound\\_theta\(\)](#), [mappel::PointEmitterModel::bounded\\_theta\(\)](#), [mappel::PointEmitterModel::bounded\\_theta\\_stack\(\)](#), [mappel::PointEmitterModel::reflected\\_theta\(\)](#), [mappel::PointEmitterModel::reflected\\_theta\\_stack\(\)](#), [mappel::PointEmitterModel::theta\\_in\\_bounds\(\)](#), and [mappel::PointEmitterModel::theta\\_stack\\_in\\_bounds\(\)](#).

**9.29.4.5** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

**9.29.4.6** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

**9.29.4.7** `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

**9.29.4.8** `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline], [inherited]

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.29.4.9** `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` [inline], [inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.29.4.10** `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` [inline], [inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.29.4.11** `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.29.4.12** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.29.4.13** `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases ( ) const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

**9.29.4.14** `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

**9.29.4.15** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline]`, `[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

**9.29.4.16** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline]`, `[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.29.4.17** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const` `[inline]`, `[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

#### 9.29.4.18 CompositeDist & mappel::PointEmitterModel::get\_prior ( ) [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

#### 9.29.4.19 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

#### 9.29.4.20 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( ) [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

#### 9.29.4.21 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( ) [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

#### 9.29.4.22 StatsT mappel::MCMCAdaptor2D::get\_stats ( ) const [protected]

Definition at line 51 of file MCMCAdaptor2D.cpp.

References eta\_y, and mappel::MCMCAdaptor1D::get\_stats().

Referenced by mappel::MCMCAdaptor2Ds::get\_stats(), and mappel::Gauss2DModel::get\_stats().

#### 9.29.4.23 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

#### 9.29.4.24 bool mappel::PointEmitterModel::has\_hyperparam ( const std::string & name ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.29.4.25** `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( ) const` `[inline],[inherited]`

Definition at line 164 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

**9.29.4.26** `template<class FillT> PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const` `[inherited]`

Definition at line 181 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.29.4.27** `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline],[inherited]`

Definition at line 172 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.29.4.28** `template<class FillT> MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.29.4.29** `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 176 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.29.4.30** `template<class FillT> CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 196 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

**9.29.4.31** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline],[inherited]`

Definition at line 168 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.29.4.32** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const [inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.29.4.33** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_I, double kappa = default_intensity_kappa ) [static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.29.4.34** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos ) [static],[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

**9.29.4.35** `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos ) [static],[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.29.4.36** `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma ) [static],[inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.



**9.29.4.37** **MCMCAdaptor2D & mappel::MCMCAdaptor2D::operator= ( const MCMCAdaptor2D & o )** [protected]

Definition at line 32 of file MCMCAdaptor2D.cpp.

References eta\_y, and mappel::MCMCAdaptor1D::operator=().

Referenced by mappel::MCMCAdaptor2Ds::operator=(), and mappel::Gauss2DModel::operator=().

**9.29.4.38** **MCMCAdaptor2D & mappel::MCMCAdaptor2D::operator= ( MCMCAdaptor2D && o )** [protected]

Definition at line 41 of file MCMCAdaptor2D.cpp.

References eta\_y, and mappel::MCMCAdaptor1D::operator=().

**9.29.4.39** **PointEmitterModel::ParamT mappel::PointEmitterModel::reflected\_theta ( const ParamT & theta ) const** [inherited]

Definition at line 276 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::reflected\_theta\_stack().

**9.29.4.40** **PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected\_theta\_stack ( const ParamVecT & theta ) const** [inherited]

Definition at line 316 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::make\_param\_stack(), and mappel::PointEmitterModel::reflected\_theta().

**9.29.4.41** **void mappel::PointEmitterModel::rename\_hyperparam ( const std::string & old\_name, const std::string & new\_name )** [inline], [inherited]

Definition at line 244 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.29.4.42** **void mappel::MCMCAdaptor2D::sample\_mcmc\_candidate ( IdxT sample\_index, ParamT & candidate, double step\_scale = 1.0 ) const**

Definition at line 59 of file MCMCAdaptor2D.cpp.

References mappel::MCMCAdaptor1D::eta\_bg, mappel::MCMCAdaptor1D::eta\_l, mappel::MCMCAdaptor1D::eta\_x, eta\_y, mappel::MCMCAdaptorBase::num\_phases, and mappel::rng\_manager.

**9.29.4.43** `void mappel::MCMCAdaptor2D::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const`

Definition at line 74 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_x`, `eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.29.4.44** `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const` `[inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

**9.29.4.45** `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline]`, `[inherited]`

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

**9.29.4.46** `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling ( double eta_bg = -1 )` `[inherited]`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

**9.29.4.47** `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )` `[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

**9.29.4.48** `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline]`, `[inherited]`

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.29.4.49 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.29.4.50 `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams ) [inline], [inherited]`

Definition at line 220 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::set\_hyperparams(), and mappel::Gauss2DsModel::set\_hyperparams().

9.29.4.51 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_l = -1 ) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_mean\_l, mappel::MCMCAdaptor1D::eta\_l, mappel::PointEmitterModel::get\_hyperparam\_value(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

9.29.4.52 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound ) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.29.4.53 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases ) [protected], [inherited]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

Referenced by mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), and mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds().

9.29.4.54 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale ) [inherited]`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::global\_max\_mcmc\_sigma\_scale, and mappel::MCMCAdaptorBase::sigma\_scale.

**9.29.4.55** `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc ) [inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.29.4.56** `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ ) [inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::set\_prior(), and mappel::Gauss2DsModel::set\_prior().

**9.29.4.57** `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ ) [inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

**9.29.4.58** `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.29.4.59** `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set\_max\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

**9.29.4.60** `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::make\_stencil(), mappel::Gauss1DsModel::make\_stencil(), mappel::Gauss2DModel::make\_stencil(), mappel::Gauss2DsxyModel::make\_stencil(), mappel::Gauss2DsModel::make\_stencil(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

9.29.4.61 **BoolVecT** mappel::PointEmitterModel::theta\_stack\_in\_bounds ( const ParamVecT & *theta* ) const [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::theta\_in\_bounds().

## 9.29.5 Member Data Documentation

9.29.5.1 **const double** mappel::PointEmitterModel::bounds\_epsilon = 1.0E-6 [static],[inherited]

Distance from the boundary to constrain in bound\_theta and bounded\_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss2↔DsModel::set\_max\_sigma\_ratio(), and mappel::PointEmitterModel::set\_ubound().

9.29.5.2 **const double** mappel::PointEmitterModel::default\_alpha\_sigma = 2 [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.29.5.3 **const double** mappel::PointEmitterModel::default\_beta\_pos = 3 [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.29.5.4 **const double** mappel::PointEmitterModel::default\_intensity\_kappa = 2 [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

9.29.5.5 **const double** mappel::PointEmitterModel::default\_max\_l = INFINITY [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::make\_prior\_component\_intensity().

9.29.5.6 **const double** mappel::PointEmitterModel::default\_mean\_l = 300 [static],[inherited]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.29.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.29.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.29.5.9** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.29.5.10** `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.29.5.11** `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.29.5.12** `double mappel::MCMCAdaptor1D::eta_x=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

**9.29.5.13** `double mappel::MCMCAdaptor2D::eta_y=0` `[protected]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor2D.h`.

Referenced by `get_stats()`, `MCMCAdaptor2D()`, `operator=()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `sample_mcmc_candidate()`.

**9.29.5.14** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static],[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

**9.29.5.15** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static],[inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.29.5.16** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static],[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.29.5.17** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static],[inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

#### 9.29.5.18 ParamT mappel::PointEmitterModel::lbound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.29.5.19 IdxT mappel::PointEmitterModel::num\_hyperparams [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

#### 9.29.5.20 IdxT mappel::PointEmitterModel::num\_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

#### 9.29.5.21 IdxT mappel::MCMCAdaptorBase::num\_phases [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1Ds::sample\_mcmc\_candidate(), mappel::MCMCAdaptor2Ds::sample\_mcmc\_candidate(), sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().

#### 9.29.5.22 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().



#### 9.29.5.23 double mappel::MCMCAdaptorBase::sigma\_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale(), mappel::MCMCAdaptorBase::get\_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), MCMCAdaptor2D::D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling(), and mappel::MCMCAdaptorBase::set\_mcmc\_sigma\_scale().

#### 9.29.5.24 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

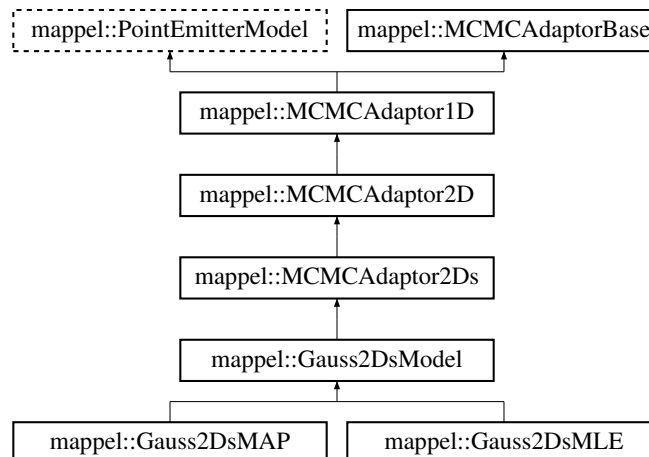
The documentation for this class was generated from the following files:

- [MCMCAdaptor2D.h](#)
- [MCMCAdaptor2D.cpp](#)

## 9.30 mappel::MCMCAdaptor2Ds Class Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/MCMCAdaptor2Ds.h>
```

Inheritance diagram for mappel::MCMCAdaptor2Ds:



## Public Types

- using `ParamT` = `arma::vec`
- using `ParamVecT` = `arma::mat`

## Public Member Functions

- void `sample_mcmc_candidate` (`IdxT` sample\_index, `ParamT` &candidate, double step\_scale=1.0) const
- void `sample_mcmc_candidate` (`IdxT` sample\_index, `ParamT` &candidate, const `IdxVecT` &fixed\_parameters\_↵ mask, double step\_scale=1.0) const
- void `set_intensity_mcmc_sampling` (double eta\_l=-1)
- void `set_background_mcmc_sampling` (double eta\_bg=-1)
- `IdxT` `get_num_params` () const
- void `check_param_shape` (const `ParamT` &theta) const
- void `check_param_shape` (const `ParamVecT` &theta) const
- void `check_psf_sigma` (double psf\_sigma) const
- void `check_psf_sigma` (const `VecT` &psf\_sigma) const
- `ParamT` `make_param` () const
- template<class FillT >  
  `ParamT` `make_param` (FillT fill) const
- `ParamVecT` `make_param_stack` (`IdxT` n) const
- template<class FillT >  
  `ParamVecT` `make_param_stack` (`IdxT` n, FillT fill) const
- `MatT` `make_param_mat` () const
- template<class FillT >  
  `MatT` `make_param_mat` (FillT fill) const
- `CubeT` `make_param_mat_stack` (`IdxT` n) const
- template<class FillT >  
  `CubeT` `make_param_mat_stack` (`IdxT` n, FillT fill) const
- CompositeDist & `get_prior` ()
- const CompositeDist & `get_prior` () const
- void `set_prior` (CompositeDist &&prior\_)
- void `set_prior` (const CompositeDist &prior\_)
- `IdxT` `get_num_hyperparams` () const
- void `set_hyperparams` (const `VecT` &hyperparams)
- `VecT` `get_hyperparams` () const
- bool `has_hyperparam` (const std::string &name) const
- double `get_hyperparam_value` (const std::string &name) const
- int `get_hyperparam_index` (const std::string &name) const
- void `set_hyperparam_value` (const std::string &name, double value)
- void `rename_hyperparam` (const std::string &old\_name, const std::string &new\_name)
- `StringVecT` `get_param_names` () const
- void `set_param_names` (const `StringVecT` &desc)
- `StringVecT` `get_hyperparam_names` () const
- void `set_hyperparam_names` (const `StringVecT` &desc)
- template<class RngT >  
  `ParamT` `sample_prior` (RngT &rng) const
- `ParamT` `sample_prior` () const
- void `set_bounds` (const `ParamT` &lbound, const `ParamT` &ubound)
- void `set_lbound` (const `ParamT` &lbound)

- void `set_ubound` (const `ParamT` &`ubound`)
- const `ParamT` & `get_lbound` () const
- const `ParamT` & `get_ubound` () const
- bool `theta_in_bounds` (const `ParamT` &`theta`) const
- void `bound_theta` (`ParamT` &`theta`, double `epsilon=bounds_epsilon`) const
- `ParamT` `bounded_theta` (const `ParamT` &`theta`, double `epsilon=bounds_epsilon`) const
- `ParamT` `reflected_theta` (const `ParamT` &`theta`) const
- `BoolVecT` `theta_stack_in_bounds` (const `ParamVecT` &`theta`) const
- `ParamVecT` `bounded_theta_stack` (const `ParamVecT` &`theta`, double `epsilon=bounds_epsilon`) const
- `ParamVecT` `reflected_theta_stack` (const `ParamVecT` &`theta`) const
- void `set_mcmc_sigma_scale` (double `scale`)
- double `get_mcmc_sigma_scale` () const
- `IdxT` `get_mcmc_num_phases` () const

#### Static Public Member Functions

- static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` `size`, double `pos_↵` `sigma=default_sigma_pos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` `size`, double `pos_↵` `beta=default_beta_pos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double `mean=default_mean_↵` `l`, double `kappa=default_intensity_kappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double `min_sigma`, double `max_↵` `sigma`, double `alpha=default_alpha_sigma`)
- static void `set_rng_seed` (`RngSeedT` `seed`)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

#### Static Public Attributes

- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1
- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2
- static const double `global_default_mcmc_sigma_scale` = 0.05
- static const double `global_max_mcmc_sigma_scale` = 0.5

### Protected Member Functions

- [MCMCAdaptor2Ds \(\)](#)
- [MCMCAdaptor2Ds \(double \[sigma\\\_scale\]\(#\)\)](#)
- [MCMCAdaptor2Ds \(const \[MCMCAdaptor2Ds\]\(#\) &o\)](#)
- [MCMCAdaptor2Ds \(\[MCMCAdaptor2Ds\]\(#\) &&o\)](#)
- [MCMCAdaptor2Ds & operator= \(const \[MCMCAdaptor2Ds\]\(#\) &o\)](#)
- [MCMCAdaptor2Ds & operator= \(\[MCMCAdaptor2Ds\]\(#\) &&o\)](#)
- [StatsT \[get\\\_stats\]\(#\) \(\) const](#)
- [void \[set\\\_mcmc\\\_num\\\_phases\]\(#\) \(\[IdxT\]\(#\) \[num\\\_phases\]\(#\)\)](#)

### Protected Attributes

- double [eta\\_sigma](#) =0
- double [eta\\_y](#) =0
- double [eta\\_x](#) =0
- double [eta\\_l](#) =0
- double [eta\\_bg](#) =0
- CompositeDist [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [IdxT](#) [num\\_phases](#)
- double [sigma\\_scale](#)

#### 9.30.1 Detailed Description

Definition at line 14 of file [MCMCAdaptor2Ds.h](#).

#### 9.30.2 Member Typedef Documentation

##### 9.30.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [\[inherited\]](#)

Parameter vector

Definition at line 47 of file [PointEmitterModel.h](#).

##### 9.30.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [\[inherited\]](#)

Vector of parameter vectors

Definition at line 48 of file [PointEmitterModel.h](#).

## 9.30.3 Constructor &amp; Destructor Documentation

## 9.30.3.1 mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds ( ) [protected]

Definition at line 11 of file MCMCAdaptor2Ds.cpp.

References mappel::MCMCAdaptorBase::global\_default\_mcmc\_sigma\_scale.

9.30.3.2 mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds ( double *sigma\_scale* ) [explicit],[protected]

Definition at line 14 of file MCMCAdaptor2Ds.cpp.

References eta\_sigma, mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases(), and mappel::MCMCAdaptorBase::sigma\_scale.

## 9.30.3.3 mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds ( const MCMCAdaptor2Ds &amp; o ) [protected]

Definition at line 22 of file MCMCAdaptor2Ds.cpp.

References eta\_sigma.

## 9.30.3.4 mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds ( MCMCAdaptor2Ds &amp;&amp; o ) [protected]

Definition at line 27 of file MCMCAdaptor2Ds.cpp.

References eta\_sigma.

## 9.30.4 Member Function Documentation

9.30.4.1 void mappel::PointEmitterModel::bound\_theta ( ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

9.30.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

**9.30.4.3** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack ( const ParamVecT & theta, double epsilon = bounds_epsilon ) const` [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

**9.30.4.4** `void mappel::PointEmitterModel::check_param_shape ( const ParamT & theta ) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

**9.30.4.5** `void mappel::PointEmitterModel::check_param_shape ( const ParamVecT & theta ) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

**9.30.4.6** `void mappel::PointEmitterModel::check_psf_sigma ( double psf_sigma ) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

**9.30.4.7** `void mappel::PointEmitterModel::check_psf_sigma ( const VecT & psf_sigma ) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

**9.30.4.8** `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` [inline], [inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.30.4.9 **StringVecT** mappel::PointEmitterModel::get\_hyperparam\_names ( ) const [inline],[inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.30.4.10 **double** mappel::PointEmitterModel::get\_hyperparam\_value ( const std::string & name ) const [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

9.30.4.11 **PointEmitterModel::ParamT** mappel::PointEmitterModel::get\_hyperparams ( ) const [inline],[inherited]

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.30.4.12 **const PointEmitterModel::ParamT &** mappel::PointEmitterModel::get\_lbound ( ) const [inline],[inherited]

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

9.30.4.13 **IdxT** mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases ( ) const [inherited]

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num\_phases.

9.30.4.14 **double** mappel::MCMCAdaptorBase::get\_mcmc\_sigma\_scale ( ) const [inherited]

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma\_scale.

9.30.4.15 **IdxT** mappel::PointEmitterModel::get\_num\_hyperparams ( ) const [inline],[inherited]

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

**9.30.4.16** `IdxT mappel::PointEmitterModel::get_num_params ( ) const [inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.30.4.17** `StringVecT mappel::PointEmitterModel::get_param_names ( ) const [inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.30.4.18** `CompositeDist & mappel::PointEmitterModel::get_prior ( ) [inline],[inherited]`

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.30.4.19** `const CompositeDist & mappel::PointEmitterModel::get_prior ( ) const [inline],[inherited]`

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.30.4.20** `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ( ) [static],[inherited]`

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.30.4.21** `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ( ) [static],[inherited]`

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.30.4.22** `StatsT mappel::MCMCAdaptor2Ds::get_stats ( ) const [protected]`

Definition at line 48 of file MCMCAdaptor2Ds.cpp.

References eta\_sigma, and mappel::MCMCAdaptor2D::get\_stats().

Referenced by mappel::Gauss2DsModel::get\_stats().



**9.30.4.23** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

**9.30.4.24** `bool mappel::PointEmitterModel::has_hyperparam ( const std::string & name ) const` `[inline]`,`[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.30.4.25** `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( ) const` `[inline]`,`[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

**9.30.4.26** `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const`  
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.30.4.27** `MatT mappel::PointEmitterModel::make_param_mat ( ) const` `[inline]`,`[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.30.4.28** `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.30.4.29 **CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n* ) const [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.30.4.30 **template<class FillT > CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n*, FillT *fill* ) const [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.30.4.31 **PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make\_param\_stack ( IdxT *n* ) const [inline],[inherited]

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack(), and mappel::PointEmitterModel::reflected\_theta\_stack().

9.30.4.32 **template<class FillT > PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make\_param\_stack ( IdxT *n*, FillT *fill* ) const [inherited]

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.30.4.33 **prior\_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make\_prior\_component\_intensity ( double *mean* = default\_mean\_I, double *kappa* = default\_intensity\_kappa ) [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default\_max\_I.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss1DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.30.4.34 **prior\_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make\_prior\_component\_position\_beta ( *IdxT size*, *double pos\_beta = default\_beta\_pos* ) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss1DModel::make\_prior\_beta\_position(), and mappel::Gauss2DsModel::make\_prior\_beta\_position().

9.30.4.35 **prior\_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make\_prior\_component\_position\_normal ( *IdxT size*, *double pos\_sigma = default\_sigma\_pos* ) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.30.4.36 **prior\_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make\_prior\_component\_sigma ( *double min\_sigma*, *double max\_sigma*, *double alpha = default\_alpha\_sigma* ) [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::Gauss1DsModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss1DsModel::make\_prior\_normal\_position(), and mappel::Gauss2DsModel::make\_prior\_normal\_position().

9.30.4.37 **MCMCAdaptor2Ds & mappel::MCMCAdaptor2Ds::operator=** ( *const MCMCAdaptor2Ds & o* ) [protected]

Definition at line 32 of file MCMCAdaptor2Ds.cpp.

References eta\_sigma, and mappel::MCMCAdaptor2D::operator=().

Referenced by mappel::Gauss2DsModel::operator=().

9.30.4.38 **MCMCAdaptor2Ds & mappel::MCMCAdaptor2Ds::operator=** ( *MCMCAdaptor2Ds && o* ) [protected]

Definition at line 40 of file MCMCAdaptor2Ds.cpp.

References eta\_sigma, and mappel::MCMCAdaptor2D::operator=().

**9.30.4.39** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`  
`[inherited]`

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.30.4.40** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const`  
`[inherited]`

Definition at line 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.30.4.41** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )`  
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

**9.30.4.42** `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, double step_scale = 1.0 ) const`

Definition at line 56 of file `MCMCAdaptor2Ds.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.30.4.43** `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate ( IdxT sample_index, ParamT & candidate, const IdxVecT & fixed_parameters_mask, double step_scale = 1.0 ) const`

Definition at line 75 of file `MCMCAdaptor2Ds.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `eta_sigma`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

**9.30.4.44** `template<class RngT> PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.30.4.45 **PointEmitterModel::ParamT** mappel::PointEmitterModel::sample\_prior ( ) const [inline],[inherited]

Definition at line 268 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior, and mappel::rng\_manager.

9.30.4.46 **void** mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling ( double *eta\_bg* = -1 ) [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default\_pixel\_mean\_bg, mappel::MCMCAdaptor1D::eta\_bg, mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_ubound(), and mappel::MCMCAdaptorBase::sigma\_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

9.30.4.47 **void** mappel::PointEmitterModel::set\_bounds ( const ParamT & *lbound\_*, const ParamT & *ubound\_* ) [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds\_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

9.30.4.48 **void** mappel::PointEmitterModel::set\_hyperparam\_names ( const StringVecT & *desc* ) [inline],[inherited]

Definition at line 260 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.30.4.49 **void** mappel::PointEmitterModel::set\_hyperparam\_value ( const std::string & *name*, double *value* ) [inline],[inherited]

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.30.4.50 **void** mappel::PointEmitterModel::set\_hyperparams ( const VecT & *hyperparams* ) [inline],[inherited]

Definition at line 220 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::set\_hyperparams(), and mappel::Gauss2DsModel::set\_hyperparams().

9.30.4.51 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling ( double eta_l = -1 )` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_l`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

9.30.4.52 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.30.4.53 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` [protected],[inherited]

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `MCMCAdaptor2Ds()`.

9.30.4.54 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )` [inherited]

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale`, and `mappel::MCMCAdaptorBase::sigma_scale`.

9.30.4.55 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.30.4.56 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )` [inherited]

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

9.30.4.57 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ ) [inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.30.4.58 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.30.4.59 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound ) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

9.30.4.60 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxyModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

9.30.4.61 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const [inherited]`

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

## 9.30.5 Member Data Documentation

9.30.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static],[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

**9.30.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

**9.30.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

**9.30.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.30.5.5** `const double mappel::PointEmitterModel::default_max_I = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.30.5.6** `const double mappel::PointEmitterModel::default_mean_I = 300` [static],[inherited]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.30.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` [static],[inherited]

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.



9.30.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

9.30.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.30.5.10 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 33 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.30.5.11 `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 32 of file MCMCAdaptor1D.h.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.30.5.12 `double mappel::MCMCAdaptor2Ds::eta_sigma = 0` `[protected]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 27 of file MCMCAdaptor2Ds.h.

Referenced by `get_stats()`, `MCMCAdaptor2Ds()`, `operator=()`, and `sample_mcmc_candidate()`.

**9.30.5.13** `double mappel::MCMCAdaptor1D::eta_x=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 31 of file `MCMCAdaptor1D.h`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1D::operator=()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`.

**9.30.5.14** `double mappel::MCMCAdaptor2D::eta_y=0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 28 of file `MCMCAdaptor2D.h`.

Referenced by `mappel::MCMCAdaptor2D::get_stats()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2D::operator=()`, `sample_mcmc_candidate()`, and `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`.

**9.30.5.15** `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static],[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `MCMCAdaptor2Ds()`.

**9.30.5.16** `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static],[inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

**9.30.5.17** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static],[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.30.5.18** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static],[inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.30.5.19 ParamT mappel::PointEmitterModel::lbound** [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_lbound(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

**9.30.5.20 IdxT mappel::PointEmitterModel::num\_hyperparams** [protected],[inherited]

Definition at line 147 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get\_num\_hyperparams(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set\_prior().

**9.30.5.21 IdxT mappel::PointEmitterModel::num\_params** [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::get\_num\_params(), mappel::PointEmitterModel::get\_stats(), mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::make\_param(), mappel::PointEmitterModel::make\_param\_mat(), mappel::PointEmitterModel::make\_param\_mat\_stack(), mappel::PointEmitterModel::make\_param\_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

**9.30.5.22 IdxT mappel::MCMCAdaptorBase::num\_phases** [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases(), mappel::MCMCAdaptorBase::get\_stats(), sample\_mcmc\_candidate(), mappel::MCMCAdaptor2D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), mappel::MCMCAdaptor1D::sample\_mcmc\_candidate(), and mappel::MCMCAdaptorBase::set\_mcmc\_num\_phases().

**9.30.5.23 CompositeDist mappel::PointEmitterModel::prior** [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DsModel::debug\_internal\_sum\_model\_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get\_hyperparam\_index(), mappel::PointEmitterModel::get\_hyperparam\_names(), mappel::PointEmitterModel::get\_hyperparam\_value(), mappel::PointEmitterModel::get\_hyperparams(), mappel::Gauss1DsModel::get\_max\_sigma(), mappel::Gauss1DsModel::get\_min\_sigma(), mappel::PointEmitterModel::get\_param\_names(), mappel::PointEmitterModel::get\_prior(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::has\_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename\_hyperparam(), mappel::PointEmitterModel::sample\_prior(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_hyperparam\_names(), mappel::PointEmitterModel::set\_hyperparam\_value(), mappel::PointEmitterModel::set\_hyperparams(), mappel::PointEmitterModel::set\_lbound(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::PointEmitterModel::set\_param\_names(), mappel::PointEmitterModel::set\_prior(), and mappel::PointEmitterModel::set\_ubound().

#### 9.30.5.24 `double mappel::MCMCAdaptorBase::sigma_scale` [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

#### 9.30.5.25 `ParamT mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

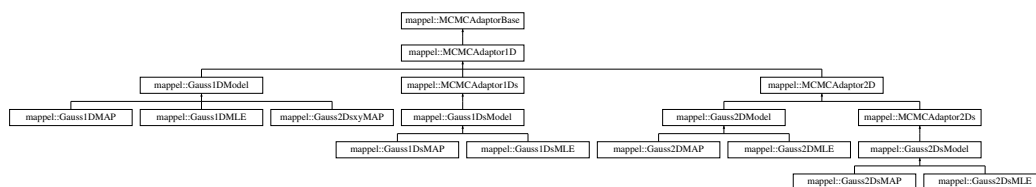
The documentation for this class was generated from the following files:

- [MCMCAdaptor2Ds.h](#)
- [MCMCAdaptor2Ds.cpp](#)

## 9.31 `mappel::MCMCAdaptorBase` Class Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/MCMCAdaptorBase.h>
```

Inheritance diagram for `mappel::MCMCAdaptorBase`:



### Public Member Functions

- void [set\\_mcmc\\_sigma\\_scale](#) (double scale)
- double [get\\_mcmc\\_sigma\\_scale](#) () const
- `IdxT` [get\\_mcmc\\_num\\_phases](#) () const

#### Static Public Attributes

- static const double [global\\_default\\_mcmc\\_sigma\\_scale](#) = 0.05
- static const double [global\\_max\\_mcmc\\_sigma\\_scale](#) = 0.5

#### Protected Member Functions

- [MCMCAdaptorBase](#) ([IdxT num\\_phases](#))
- [MCMCAdaptorBase](#) ([IdxT num\\_phases](#), double [sigma\\_scale](#))
- void [set\\_mcmc\\_num\\_phases](#) ([IdxT num\\_phases](#))
- [StatsT get\\_stats](#) () const

#### Protected Attributes

- [IdxT num\\_phases](#)
- double [sigma\\_scale](#)

#### 9.31.1 Detailed Description

Definition at line 13 of file [MCMCAdaptorBase.h](#).

#### 9.31.2 Constructor & Destructor Documentation

##### 9.31.2.1 mappel::MCMCAdaptorBase::MCMCAdaptorBase ( [IdxT num\\_phases](#) ) [protected]

Definition at line 14 of file [MCMCAdaptorBase.cpp](#).

##### 9.31.2.2 mappel::MCMCAdaptorBase::MCMCAdaptorBase ( [IdxT num\\_phases](#), double [sigma\\_scale](#) ) [protected]

Definition at line 18 of file [MCMCAdaptorBase.cpp](#).

References [global\\_max\\_mcmc\\_sigma\\_scale](#).

#### 9.31.3 Member Function Documentation

##### 9.31.3.1 [IdxT](#) mappel::MCMCAdaptorBase::get\_mcmc\_num\_phases ( ) const

Definition at line 56 of file [MCMCAdaptorBase.cpp](#).

References [num\\_phases](#).

### 9.31.3.2 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale ( ) const`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `sigma_scale`.

### 9.31.3.3 `StatsT mappel::MCMCAdaptorBase::get_stats ( ) const` `[protected]`

Definition at line 70 of file MCMCAdaptorBase.cpp.

References `num_phases`, and `sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::get_stats()`.

### 9.31.3.4 `void mappel::MCMCAdaptorBase::set_mcmc_num_phases ( IdxT num_phases )` `[protected]`

Definition at line 59 of file MCMCAdaptorBase.cpp.

References `num_phases`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

### 9.31.3.5 `void mappel::MCMCAdaptorBase::set_mcmc_sigma_scale ( double scale )`

Definition at line 39 of file MCMCAdaptorBase.cpp.

References `global_max_mcmc_sigma_scale`, and `sigma_scale`.

## 9.31.4 Member Data Documentation

### 9.31.4.1 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

### 9.31.4.2 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `MCMCAdaptorBase()`, and `set_mcmc_sigma_scale()`.

9.31.4.3 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file `MCMCAdaptorBase.h`.

Referenced by `get_mcmc_num_phases()`, `get_stats()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `set_mcmc_num_phases()`.

9.31.4.4 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file `MCMCAdaptorBase.h`.

Referenced by `get_mcmc_sigma_scale()`, `get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `set_mcmc_sigma_scale()`.

The documentation for this class was generated from the following files:

- [MCMCAdaptorBase.h](#)
- [MCMCAdaptorBase.cpp](#)

## 9.32 mappel::mcmc::MCMCData Struct Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/mcmc_data.h>
```

## Public Member Functions

- void [initialize\\_arrays](#) ([IdxT](#) Nparams)

## Public Attributes

- [IdxT](#) Nsample =0
- [IdxT](#) Nburnin =0
- [IdxT](#) thin =0
- double [confidence](#) =-INFINITY
- [VecT](#) [sample\\_mean](#)
- [MatT](#) [sample\\_cov](#)
- [VecT](#) [credible\\_lb](#)
- [VecT](#) [credible\\_ub](#)
- [MatT](#) [sample](#)
- [VecT](#) [sample\\_rllh](#)

### 9.32.1 Detailed Description

Structures for reporting MCMC resultsData and controlling parameters for an MCMC posterior sampling for a single data.

Definition at line 21 of file mcmc\_data.h.

### 9.32.2 Member Function Documentation

#### 9.32.2.1 void mappel::mcmc::MCMCData::initialize\_arrays ( IdxT Nparams )

Definition at line 12 of file mcmc.cpp.

References credible\_lb, credible\_ub, Nsample, sample, sample\_cov, sample\_mean, and sample\_rllh.

### 9.32.3 Member Data Documentation

#### 9.32.3.1 double mappel::mcmc::MCMCData::confidence =-INFINITY

Definition at line 26 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior().

#### 9.32.3.2 VecT mappel::mcmc::MCMCData::credible\_lb

Definition at line 30 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior(), initialize\_arrays(), and mappel::mcmc::MCMCDataStack↵::initialize\_arrays().

#### 9.32.3.3 VecT mappel::mcmc::MCMCData::credible\_ub

Definition at line 31 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior(), initialize\_arrays(), and mappel::mcmc::MCMCDataStack↵::initialize\_arrays().

#### 9.32.3.4 IdxT mappel::mcmc::MCMCData::Nburnin =0

Definition at line 24 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior().

#### 9.32.3.5 IdxT mappel::mcmc::MCMCData::Nsample =0

Definition at line 23 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior(), initialize\_arrays(), mappel::mcmc::MCMCDebugData↵::initialize\_arrays(), and mappel::mcmc::MCMCDataStack::initialize\_arrays().



#### 9.32.3.6 MatT mappel::mcmc::MCMCData::sample

Definition at line 32 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior(), initialize\_arrays(), mappel::mcmc::MCMCDebugData::initialize\_arrays(), and mappel::mcmc::MCMCDataStack::initialize\_arrays().

#### 9.32.3.7 MatT mappel::mcmc::MCMCData::sample\_cov

Definition at line 29 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior(), initialize\_arrays(), and mappel::mcmc::MCMCDataStack::initialize\_arrays().

#### 9.32.3.8 VecT mappel::mcmc::MCMCData::sample\_mean

Definition at line 28 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior(), initialize\_arrays(), and mappel::mcmc::MCMCDataStack::initialize\_arrays().

#### 9.32.3.9 VecT mappel::mcmc::MCMCData::sample\_rlh

Definition at line 33 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior(), initialize\_arrays(), mappel::mcmc::MCMCDebugData::initialize\_arrays(), mappel::mcmc::MCMCDataStack::initialize\_arrays(), and mappel::mcmc::thin\_sample().

#### 9.32.3.10 IdxT mappel::mcmc::MCMCData::thin =0

Definition at line 25 of file mcmc\_data.h.

Referenced by mappel::methods::estimate\_posterior(), and mappel::mcmc::thin\_sample().

The documentation for this struct was generated from the following files:

- [mcmc\\_data.h](#)
- [mcmc.cpp](#)

### 9.33 mappel::mcmc::MCMCDataStack Struct Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/mcmc_data.h>
```

#### Public Member Functions

- void [initialize\\_arrays](#) (IdxT Nparams)

## Public Attributes

- `IdxT Nsample` =0
- `IdxT Nburnin` =0
- `IdxT thin` =0
- `double confidence` =-INFINITY
- `IdxT Ndata` =0
- `MatT sample_mean`
- `CubeT sample_cov`
- `MatT credible_lb`
- `MatT credible_ub`
- `CubeT sample`
- `MatT sample_rllh`

### 9.33.1 Detailed Description

Data and controlling parameters for an MCMC posterior sampling for stack of single data.

Definition at line 53 of file `mcmc_data.h`.

### 9.33.2 Member Function Documentation

#### 9.33.2.1 `void mappel::mcmc::MCMCDataStack::initialize_arrays ( IdxT Nparams )`

Definition at line 30 of file `mcmc.cpp`.

References `mappel::mcmc::MCMCData::credible_lb`, `mappel::mcmc::MCMCData::credible_ub`, `mappel::mcmc::MCMCData::Nsample`, `mappel::mcmc::MCMCData::sample`, `mappel::mcmc::MCMCData::sample_cov`, `mappel::mcmc::MCMCData::sample_mean`, and `mappel::mcmc::MCMCData::sample_rllh`.

Referenced by `mappel::methods::openmp::estimate_posterior_stack()`.

### 9.33.3 Member Data Documentation

#### 9.33.3.1 `double mappel::mcmc::MCMCDataStack::confidence` =-INFINITY

Definition at line 58 of file `mcmc_data.h`.

#### 9.33.3.2 `MatT mappel::mcmc::MCMCDataStack::credible_lb`

Definition at line 64 of file `mcmc_data.h`.

#### 9.33.3.3 `MatT mappel::mcmc::MCMCDataStack::credible_ub`

Definition at line 65 of file `mcmc_data.h`.

**9.33.3.4 IdxT mappel::mcmc::MCMCDataStack::Nburnin =0**

Definition at line 56 of file mcmc\_data.h.

Referenced by mappel::methods::openmp::estimate\_posterior\_stack().

**9.33.3.5 IdxT mappel::mcmc::MCMCDataStack::Ndata =0**

Definition at line 61 of file mcmc\_data.h.

Referenced by mappel::methods::openmp::estimate\_posterior\_stack().

**9.33.3.6 IdxT mappel::mcmc::MCMCDataStack::Nsample =0**

Definition at line 55 of file mcmc\_data.h.

Referenced by mappel::methods::openmp::estimate\_posterior\_stack().

**9.33.3.7 CubeT mappel::mcmc::MCMCDataStack::sample**

Definition at line 66 of file mcmc\_data.h.

**9.33.3.8 CubeT mappel::mcmc::MCMCDataStack::sample\_cov**

Definition at line 63 of file mcmc\_data.h.

**9.33.3.9 MatT mappel::mcmc::MCMCDataStack::sample\_mean**

Definition at line 62 of file mcmc\_data.h.

**9.33.3.10 MatT mappel::mcmc::MCMCDataStack::sample\_rllh**

Definition at line 67 of file mcmc\_data.h.

**9.33.3.11 IdxT mappel::mcmc::MCMCDataStack::thin =0**

Definition at line 57 of file mcmc\_data.h.

Referenced by mappel::methods::openmp::estimate\_posterior\_stack().

The documentation for this struct was generated from the following files:

- [mcmc\\_data.h](#)
- [mcmc.cpp](#)

### 9.34 mappel::mcmc::MCMCDebugData Struct Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/mcmc_data.h>
```

#### Public Member Functions

- void [initialize\\_arrays](#) ([IdxT](#) Nparams)

#### Public Attributes

- [IdxT](#) Nsample =0
- [MatT](#) sample
- [VecT](#) sample\_rllh
- [MatT](#) candidate
- [VecT](#) candidate\_rllh

#### 9.34.1 Detailed Description

Debugging Data and controlling parameters for an MCMC posterior sampling for a single data. No burnin or thinning is performed when debugging

Definition at line 40 of file `mcmc_data.h`.

#### 9.34.2 Member Function Documentation

##### 9.34.2.1 void mappel::mcmc::MCMCDebugData::initialize\_arrays ( [IdxT](#) *Nparams* )

Definition at line 22 of file `mcmc.cpp`.

References `mappel::mcmc::MCMCData::Nsample`, `mappel::mcmc::MCMCData::sample`, and `mappel::mcmc::MCMCData::sample_rllh`.

Referenced by `mappel::methods::debug::estimate_posterior_debug()`.

#### 9.34.3 Member Data Documentation

##### 9.34.3.1 [MatT](#) mappel::mcmc::MCMCDebugData::candidate

Definition at line 46 of file `mcmc_data.h`.

Referenced by `mappel::methods::debug::estimate_posterior_debug()`.

## 9.34.3.2 VecT mappel::mcmc::MCMCDebugData::candidate\_rllh

Definition at line 47 of file mcmc\_data.h.

Referenced by mappel::methods::debug::estimate\_posterior\_debug().

## 9.34.3.3 IdxT mappel::mcmc::MCMCDebugData::Nsample =0

Definition at line 42 of file mcmc\_data.h.

Referenced by mappel::methods::debug::estimate\_posterior\_debug().

## 9.34.3.4 MatT mappel::mcmc::MCMCDebugData::sample

Definition at line 44 of file mcmc\_data.h.

Referenced by mappel::methods::debug::estimate\_posterior\_debug().

## 9.34.3.5 VecT mappel::mcmc::MCMCDebugData::sample\_rllh

Definition at line 45 of file mcmc\_data.h.

Referenced by mappel::methods::debug::estimate\_posterior\_debug().

The documentation for this struct was generated from the following files:

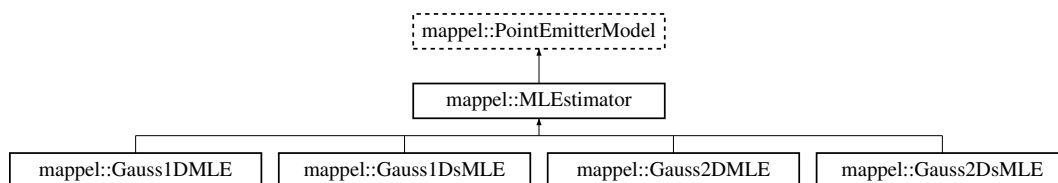
- [mcmc\\_data.h](#)
- [mcmc.cpp](#)

## 9.35 mappel::MLEstimator Class Reference

A Mixin class to configure a for MLE estimation (null prior).

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/MLEstimator.h>
```

Inheritance diagram for mappel::MLEstimator:



## Public Types

- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat

## Public Member Functions

- [StatsT get\\_stats](#) () const
- [IdxT get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT make\\_param](#) () const
- template<class FillT >  
  [ParamT make\\_param](#) (FillT fill) const
- [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT make\\_param\\_mat](#) () const
- template<class FillT >  
  [MatT make\\_param\\_mat](#) (FillT fill) const
- [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const
- template<class FillT >  
  [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get\\_prior](#) ()
- const CompositeDist & [get\\_prior](#) () const
- void [set\\_prior](#) (CompositeDist &&prior\_)
- void [set\\_prior](#) (const CompositeDist &prior\_)
- [IdxT get\\_num\\_hyperparams](#) () const
- void [set\\_hyperparams](#) (const [VecT](#) &hyperparams)
- [VecT get\\_hyperparams](#) () const
- bool [has\\_hyperparam](#) (const std::string &name) const
- double [get\\_hyperparam\\_value](#) (const std::string &name) const
- int [get\\_hyperparam\\_index](#) (const std::string &name) const
- void [set\\_hyperparam\\_value](#) (const std::string &name, double value)
- void [rename\\_hyperparam](#) (const std::string &old\_name, const std::string &new\_name)
- [StringVecT get\\_param\\_names](#) () const
- void [set\\_param\\_names](#) (const [StringVecT](#) &desc)
- [StringVecT get\\_hyperparam\\_names](#) () const
- void [set\\_hyperparam\\_names](#) (const [StringVecT](#) &desc)
- template<class RngT >  
  [ParamT sample\\_prior](#) (RngT &rng) const
- [ParamT sample\\_prior](#) () const
- void [set\\_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set\\_lbound](#) (const [ParamT](#) &lbound)
- void [set\\_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get\\_lbound](#) () const
- const [ParamT](#) & [get\\_ubound](#) () const
- bool [theta\\_in\\_bounds](#) (const [ParamT](#) &theta) const
- void [bound\\_theta](#) ([ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT bounded\\_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamT reflected\\_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta\\_stack\\_in\\_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded\\_theta\\_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds\\_epsilon](#)) const
- [ParamVecT reflected\\_theta\\_stack](#) (const [ParamVecT](#) &theta) const

## Static Public Member Functions

- static prior\_hessian::TruncatedNormalDist [make\\_prior\\_component\\_position\\_normal](#) (IdxT size, double pos\_↵ sigma=[default\\_sigma\\_pos](#))
- static prior\_hessian::ScaledSymmetricBetaDist [make\\_prior\\_component\\_position\\_beta](#) (IdxT size, double pos\_↵ beta=[default\\_beta\\_pos](#))
- static prior\_hessian::TruncatedGammaDist [make\\_prior\\_component\\_intensity](#) (double mean=[default\\_mean\\_l](#)↵ l, double kappa=[default\\_intensity\\_kappa](#))
- static prior\_hessian::TruncatedParetoDist [make\\_prior\\_component\\_sigma](#) (double min\_sigma, double max\_↵ sigma, double alpha=[default\\_alpha\\_sigma](#))
- static void [set\\_rng\\_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get\\_rng\\_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get\\_rng\\_generator](#) ()

## Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2

## Protected Member Functions

- [MLEstimator](#) ()=default
- [MLEstimator](#) (const [MLEstimator](#) &o)
- [MLEstimator](#) ([MLEstimator](#) &&o)
- [MLEstimator](#) & [operator=](#) (const [MLEstimator](#) &o)
- [MLEstimator](#) & [operator=](#) ([MLEstimator](#) &&o)

## Protected Attributes

- CompositeDist [prior](#)
- IdxT [num\\_params](#)
- IdxT [num\\_hyperparams](#)
- ParamT [lbound](#)
- ParamT [ubound](#)

### 9.35.1 Detailed Description

A Mixin class to configure a for MLE estimation (null prior).

Inheriting from this class modifies the objective function undergoing optimization to use a Null prior, by simply ignoring the effect of the prior on the objective. This which effectively turns the objective function into a pure likelihood function, and the estimator becomes an MLE estimator.

Definition at line 22 of file MLEstimator.h.

### 9.35.2 Member Typedef Documentation

#### 9.35.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

#### 9.35.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

### 9.35.3 Constructor & Destructor Documentation

#### 9.35.3.1 `mappel::MLEstimator::MLEstimator ( )` `[protected]`, `[default]`

#### 9.35.3.2 `mappel::MLEstimator::MLEstimator ( const MLEstimator & o )` `[inline]`, `[protected]`

Definition at line 25 of file MLEstimator.h.

#### 9.35.3.3 `mappel::MLEstimator::MLEstimator ( MLEstimator && o )` `[inline]`, `[protected]`

Definition at line 26 of file MLEstimator.h.

### 9.35.4 Member Function Documentation

#### 9.35.4.1 `void mappel::PointEmitterModel::bound_theta ( ParamT & theta, double epsilon = bounds_epsilon ) const` `[inherited]`

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.



**9.35.4.2 PointEmitterModel::ParamT** mappel::PointEmitterModel::bounded\_theta ( const ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check\_param\_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded\_theta\_stack().

**9.35.4.3 PointEmitterModel::ParamVecT** mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & *theta*, double *epsilon* = bounds\_epsilon ) const [inherited]

Definition at line 306 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::check\_param\_shape(), and mappel::PointEmitterModel::make\_param\_stack().

**9.35.4.4 void** mappel::PointEmitterModel::check\_param\_shape ( const ParamT & *theta* ) const [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::bounded\_theta\_stack(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::reflected\_theta\_stack(), mappel::PointEmitterModel::theta\_in\_bounds(), and mappel::PointEmitterModel::theta\_stack\_in\_bounds().

**9.35.4.5 void** mappel::PointEmitterModel::check\_param\_shape ( const ParamVecT & *theta* ) const [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num\_params.

**9.35.4.6 void** mappel::PointEmitterModel::check\_psf\_sigma ( double *psf\_sigma* ) const [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

**9.35.4.7 void** mappel::PointEmitterModel::check\_psf\_sigma ( const VecT & *psf\_sigma* ) const [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global\_max\_psf\_sigma, and mappel::PointEmitterModel::global\_min\_psf\_sigma.

**9.35.4.8** `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.35.4.9** `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const` `[inline]`,`[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.35.4.10** `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const` `[inline]`,  
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::MCMCAdaptor1D::set\_intensity\_mcmc\_sampling().

**9.35.4.11** `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const` `[inline]`,  
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.35.4.12** `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const` `[inline]`,  
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling().

**9.35.4.13** `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const` `[inline]`,`[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_hyperparams.

**9.35.4.14** `IdxT mappel::PointEmitterModel::get_num_params ( ) const` `[inline]`,`[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.35.4.15 StringVecT mappel::PointEmitterModel::get\_param\_names ( ) const** [inline],[inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.35.4.16 CompositeDist & mappel::PointEmitterModel::get\_prior ( )** [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

**9.35.4.17 const CompositeDist & mappel::PointEmitterModel::get\_prior ( ) const** [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

**9.35.4.18 ParallelRngGeneratorT & mappel::PointEmitterModel::get\_rng\_generator ( )** [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.35.4.19 ParallelRngManagerT & mappel::PointEmitterModel::get\_rng\_manager ( )** [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

**9.35.4.20 StatsT mappel::PointEmitterModel::get\_stats ( ) const** [inherited]

Definition at line 125 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num\_hyperparams, mappel::PointEmitterModel::num\_params, mappel::PointEmitterModel::prior, mappel::rng\_manager, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DModel::get\_stats(), mappel::Gauss1DsModel::get\_stats(), mappel::Gauss2DModel::get\_stats(), and mappel::Gauss2DsModel::get\_stats().

**9.35.4.21 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get\_ubound ( ) const** [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get\_max\_sigma\_ratio(), mappel::Gauss2DsModel::get\_max\_sigma\_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set\_background\_mcmc\_sampling(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.35.4.22 **bool** mappel::PointEmitterModel::has\_hyperparam ( const std::string & *name* ) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.35.4.23 **PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( ) const [inline],[inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

9.35.4.24 **template<class FillT > PointEmitterModel::ParamT** mappel::PointEmitterModel::make\_param ( FillT *fill* ) const [inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.35.4.25 **MatT** mappel::PointEmitterModel::make\_param\_mat ( ) const [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.35.4.26 **template<class FillT > MatT** mappel::PointEmitterModel::make\_param\_mat ( FillT *fill* ) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.35.4.27 **CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n* ) const [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

9.35.4.28 **template<class FillT > CubeT** mappel::PointEmitterModel::make\_param\_mat\_stack ( IdxT *n*, FillT *fill* ) const [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num\_params.

**9.35.4.29** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline]`, `[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.35.4.30** `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

**9.35.4.31** `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_I, double kappa = default_intensity_kappa )` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.35.4.32** `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos )` `[static]`, `[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

**9.35.4.33** `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos )` `[static]`, `[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.35.4.34** `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma )` `[static], [inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

**9.35.4.35** `MLEstimator& mappel::MLEstimator::operator= ( const MLEstimator & o )` `[inline], [protected]`

Definition at line 27 of file MLEstimator.h.

Referenced by `mappel::Gauss1DMLE::operator=()`, `mappel::Gauss1DsMLE::operator=()`, `mappel::Gauss2DsMLE::operator=()`, and `mappel::Gauss2DMLE::operator=()`.

**9.35.4.36** `MLEstimator& mappel::MLEstimator::operator= ( MLEstimator && o )` `[inline], [protected]`

Definition at line 28 of file MLEstimator.h.

**9.35.4.37** `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const` `[inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

**9.35.4.38** `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const` `[inherited]`

Definition at line 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

**9.35.4.39** `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name )` `[inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.35.4.40 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`  
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.35.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const` `[inline],[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.35.4.42 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`  
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.35.4.43 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc )` `[inline],[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.35.4.44 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value )` `[inline],[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.35.4.45 `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams )` `[inline],[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

9.35.4.46 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

9.35.4.47 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc )` [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.35.4.48 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )` [inherited]

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

9.35.4.49 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ )` [inherited]

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.35.4.50 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed )` [static],[inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.35.4.51 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )` [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.



9.35.4.52 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const` [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

9.35.4.53 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const` [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

## 9.35.5 Member Data Documentation

9.35.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static], [inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

9.35.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

9.35.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

9.35.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static], [inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

**9.35.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

**9.35.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

**9.35.5.7** `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

**9.35.5.8** `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

**9.35.5.9** `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

**9.35.5.10** `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.35.5.11** `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

**9.35.5.12** `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.35.5.13** `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

**9.35.5.14** `IdxT mappel::PointEmitterModel::num_params` `[protected], [inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

**9.35.5.15** `CompositeDist mappel::PointEmitterModel::prior` `[protected], [inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

### 9.35.5.16 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound\_theta(), mappel::PointEmitterModel::bounded\_theta(), mappel::PointEmitterModel::get\_stats(), mappel::PointEmitterModel::get\_ubound(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected\_theta(), mappel::PointEmitterModel::set\_bounds(), mappel::PointEmitterModel::set\_lbound(), mappel::PointEmitterModel::set\_prior(), mappel::PointEmitterModel::set\_ubound(), and mappel::PointEmitterModel::theta\_in\_bounds().

The documentation for this class was generated from the following file:

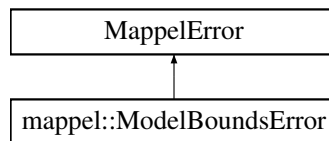
- [MLEstimator.h](#)

## 9.36 mappel::ModelBoundsError Struct Reference

Access outside the model bounds is attempted.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/util.h>
```

Inheritance diagram for mappel::ModelBoundsError:



### Public Member Functions

- [ModelBoundsError](#) (std::string message)

### 9.36.1 Detailed Description

Access outside the model bounds is attempted.

Definition at line 90 of file util.h.

### 9.36.2 Constructor & Destructor Documentation

#### 9.36.2.1 mappel::ModelBoundsError::ModelBoundsError ( std::string *message* ) [inline]

Definition at line 92 of file util.h.

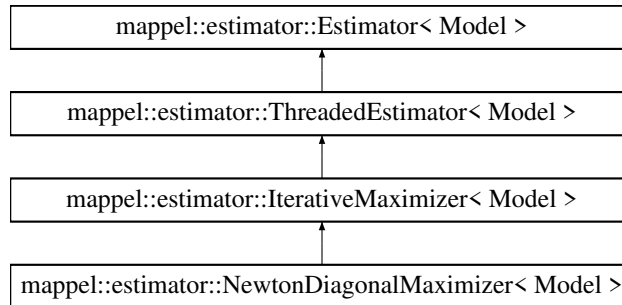
The documentation for this struct was generated from the following file:

- [util.h](#)

## 9.37 mappel::estimator::NewtonDiagonalMaximizer&lt; Model &gt; Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::estimator::NewtonDiagonalMaximizer< Model >:



## Public Types

- using [MaximizerData](#) = typename [IterativeMaximizer](#)< Model >::[MaximizerData](#)

## Public Member Functions

- [NewtonDiagonalMaximizer](#) (const Model &model, int max\_iterations=[IterativeMaximizer](#)< Model >::DefaultIterations)
- std::string name () const
- double mean\_iterations ()
- double mean\_backtracks ()
- double mean\_fun\_evals ()
- double mean\_der\_evals ()
- StatsT get\_stats ()
- StatsT get\_debug\_stats ()
- void clear\_stats ()
- int get\_total\_iterations () const
- int get\_total\_backtracks () const
- int get\_total\_fun\_evals () const
- int get\_total\_der\_evals () const
- void local\_maximize (const [ModelDataT](#)< Model > &im, [StencilT](#)< Model > &stencil, [MLEData](#) &data)  
Perform a local maximization to finish off a simulated annealing run.
- void local\_maximize (const [ModelDataT](#)< Model > &im, [StencilT](#)< Model > &stencil, [MLEDebugData](#) &debug\_data)
- void local\_profile\_maximize (const [ModelDataT](#)< Model > &im, const [IdxVecT](#) &fixed\_param\_idx, [StencilT](#)< Model > &stencil, [MLEDebugData](#) &mle)
- void estimate\_max\_stack (const [ModelDataStackT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [MLEDataStack](#) &mle\_data\_stack) override
- void estimate\_profile\_max (const [ModelDataT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [ProfileLikelihoodData](#) &profile) override
- void estimate\_profile\_bounds\_parallel (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est) override

- void `estimate_profile_bounds_stack` (const `ModelDataStackT`< `Model` > &data, `ProfileBoundsDataStack` &bounds\_est\_stack) override
- const `Model` & `get_model` ()
- void `estimate_max_stack` (const `ModelDataStackT`< `Model` > &data\_stack, `MLEDDataStack` &mle\_data\_stack)
- void `estimate_max` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDData` &mle\_data, `StencilT`< `Model` > &mle\_stencil)
- void `estimate_max` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDData` &mle\_data)
- void `estimate_max` (const `ModelDataT`< `Model` > &data, `MLEDData` &mle\_data)
- void `estimate_max_debug` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDData` &mle\_data, `StencilT`< `Model` > &mle\_stencil)
- void `estimate_max_debug` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDData` &mle\_data)
- double `estimate_profile_max` (const `ModelDataT`< `Model` > &data, const `IdxVecT` &fixed\_idx, const `ParamT`< `Model` > &fixed\_theta\_init, `StencilT`< `Model` > &theta\_max)
- void `estimate_profile_bounds` (const `ModelDataT`< `Model` > &data, `ProfileBoundsData` &bounds\_est)
- void `estimate_profile_bounds_debug` (const `ModelDataT`< `Model` > &data, `ProfileBoundsDebugData` &bounds\_est)
- `IdxVecT` `get_exit_counts` () const

#### Static Public Attributes

- static const int `DefaultIterations` =100

#### Protected Member Functions

- void `record_run_statistics` (const `MaximizerData` &data)
- void `compute_estimate` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDData` &mle\_data, `StencilT`< `Model` > &mle\_stencil) override
- void `compute_estimate_debug` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, `MLEDData` &mle\_data, `StencilT`< `Model` > &mle\_stencil) override
- double `compute_profile_estimate` (const `ModelDataT`< `Model` > &data, const `ParamT`< `Model` > &theta\_init, const `IdxVecT` &fixed\_idx, `StencilT`< `Model` > &theta\_max) override
- void `compute_profile_bound` (const `ModelDataT`< `Model` > &data, `ProfileBoundsData` &est, const `VecT` &init\_step, `IdxT` param\_idx, `IdxT` which\_bound) override
- void `compute_profile_bound_debug` (const `ModelDataT`< `Model` > &data, `ProfileBoundsDebugData` &bounds) override
- bool `backtrack` (`MaximizerData` &data)
- bool `profile_bound_backtrack` (`MaximizerData` &data, `IdxT` fixed\_idx, double target\_rllh, double old\_fval, const `VecT` &fgad)
- virtual void `maximize` (`MaximizerData` &data)=0
- virtual void `solve_profile_bound` (`MaximizerData` &data, `MLEDData` &mle, double llh\_delta, `IdxT` fixed\_idx, `IdxT` which\_bound)
- bool `convergence_test_grad_ratio` (const `VecT` &grad, double fval)
- bool `convergence_test_step_size` (const `VecT` &new\_theta, const `VecT` &old\_theta)
- void `record_exit_code` (`ExitCode` code) override
- void `record_walltime` (`ClockT`::time\_point start\_walltime, int num\_estimations)

## Protected Attributes

- int [max\\_iterations](#)
- int [total\\_iterations](#) = 0
- int [total\\_backtracks](#) = 0
- int [total\\_fun\\_evals](#) = 0
- int [total\\_der\\_evals](#) = 0
- [IdxVecT](#) [last\\_backtrack\\_idxxs](#)  
*Debugging: Stores last set of backtrack\_idxxs when data.save\_seq==true.*
- int [max\\_threads](#)
- int [num\\_threads](#)
- std::mutex [mtx](#)
- const Model & [model](#)
- int [num\\_estimations](#) = 0
- double [total\\_walltime](#) = 0.
- [IdxVecT](#) [exit\\_counts](#)

## Static Protected Attributes

- static const double [min\\_eigenvalue\\_correction\\_delta](#) = 1e-3  
*Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.*
- static const double [convergence\\_min\\_function\\_change\\_ratio](#) = 1.0e-9  
*Convergence criteria: tolerance for function-value change.*
- static const double [convergence\\_min\\_step\\_size\\_ratio](#) = 1.0e-9  
*Convergence criteria: tolerance of relative step size.*
- static const double [backtrack\\_min\\_ratio](#) = 0.05
- static const double [backtrack\\_max\\_ratio](#) = 0.50
- static const double [backtrack\\_min\\_linear\\_step\\_ratio](#) = 1e-3
- static const int [max\\_backtracks](#) = 8
- static const double [min\\_profile\\_bound\\_residual](#) = 1e-4  
*Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.*

## 9.37.1 Detailed Description

```
template<class Model>
class mappel::estimator::NewtonDiagonalMaximizer< Model >
```

Definition at line 567 of file estimator.h.

## 9.37.2 Member Typedef Documentation

9.37.2.1 `template<class Model > using mappel::estimator::NewtonDiagonalMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 570 of file estimator.h.

### 9.37.3 Constructor & Destructor Documentation

**9.37.3.1** `template<class Model> mappel::estimator::NewtonDiagonalMaximizer< Model>::NewtonDiagonalMaximizer ( const Model & model, int max_iterations = IterativeMaximizer<Model>::DefaultIterations ) [inline]`

Definition at line 572 of file estimator.h.

### 9.37.4 Member Function Documentation

**9.37.4.1** `template<class Model> bool mappel::estimator::IterativeMaximizer< Model>::backtrack ( MaximizerData & data ) [protected], [inherited]`

Definition at line 870 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::backtrack_min_linear_step_ratio`, `mappel::clamp()`, `mappel::estimator::IterativeMaximizer< Model>::convergence_min_step_size_ratio`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::grad`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::im`, `mappel::estimator::IterativeMaximizer< Model>::max_backtracks`, `mappel::estimator::MaxBacktracks`, `mappel::estimator::Estimator< Model>::model`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::saved_theta()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::set_stencil()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::step`, `mappel::estimator::StepSize`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.37.4.2** `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::clear_stats ( ) [virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model>](#).

Definition at line 848 of file estimator\_impl.h.

References `mappel::estimator::ThreadedEstimator< Model>::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model>::mtx`, `mappel::estimator::IterativeMaximizer< Model>::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model>::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model>::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model>::total_iterations`.



9.37.4.3 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil )` [override], [protected], [virtual], [inherited]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 1043 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::methods::observed\\_information\(\)](#), [mappel::estimator::MLEData::obsI](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::MLEData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), [mappel::estimator::MLEData::theta](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

9.37.4.4 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug ( const ModelDataT< Model > & im, const ParamT< Model > & theta_init, MLEDebugData & mle_debug, StencilT< Model > & mle_stencil )` [override], [protected], [virtual], [inherited]

Virtual `estimate_debug` interface

Estimators that produce a sequence of results (e.g. `IterativeEstimators`) can override this dummy debug implementation.

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1057 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get\\_theta\\_sequence\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get\\_theta\\_sequence\\_rllh\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::methods::observed\\_information\(\)](#), [mappel::estimator::MLEDebugData::obsI](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::MLEDebugData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::MLEDebugData::sequence](#), [mappel::estimator::MLEDebugData::sequence\\_rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), [mappel::estimator::MLEDebugData::theta](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

9.37.4.5 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound ( const ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound )` [override], [protected], [virtual], [inherited]

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1088 of file estimator\_impl.h.

References [mappel::estimator::ProfileBoundsData::estimated\\_idx](#), [mappel::estimator::ProfileBoundsData::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsData::profile\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_ub](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::solve\\_profile\\_bound\(\)](#), [mappel::estimator::ProfileBoundsData::target\\_rllh\\_delta](#), and [mappel::estimator::MLEData::theta](#).

**9.37.4.6** `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds ) [override], [protected], [virtual], [inherited]`

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1114 of file `estimator_impl.h`.

References `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsDebugData::Nseq_lb`, `mappel::estimator::ProfileBoundsDebugData::Nseq_ub`, `mappel::estimator::MLEData::obs1`, `mappel::estimator::ProfileBoundsDebugData::profile_lb`, `mappel::estimator::ProfileBoundsDebugData::profile_ub`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::ProfileBoundsDebugData::sequence_lb`, `mappel::estimator::ProfileBoundsDebugData::sequence_lb_rllh`, `mappel::estimator::ProfileBoundsDebugData::sequence_ub`, `mappel::estimator::ProfileBoundsDebugData::sequence_ub_rllh`, `mappel::estimator::IterativeMaximizer< Model >::solve_profile_bound()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

**9.37.4.7** `template<class Model> double mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxs, StencilT< Model > & theta_max ) [override], [protected], [virtual], [inherited]`

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1074 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`.

**9.37.4.8** `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio ( const VecT & grad, double fval ) [protected], [inherited]`

Definition at line 1015 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio`, `mappel::estimator::GradRatio`, `mappel::norm_sq()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::square()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.37.4.9** `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size ( const VecT & new_theta, const VecT & old_theta ) [protected], [inherited]`

Definition at line 1027 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::norm_sq()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::StepSize`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.37.4.10 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`.

9.37.4.11 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.

9.37.4.12 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::Estimator< Model >::model`.

9.37.4.13 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at `theta_init`, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.37.4.14 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data )`  
`[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

9.37.4.15 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack )` `[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max_stack()`, and `mappel::estimator::Estimator< Model >::model`.

9.37.4.16 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack )` [override],[virtual],[inherited]

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEDataStack::Ndata`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDataStack::obsI`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDataStack::rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDataStack::theta`.

9.37.4.17 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est )` [inherited]

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

9.37.4.18 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est )` [inherited]

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

9.37.4.19 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est )` [override], [virtual], [inherited]

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.

9.37.4.20 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est )` [override], [virtual], [inherited]

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, and `mappel::estimator::ProfileBoundsDataStack::initialize_arrays()`.

mappel::estimator::ProfileBoundsData::mle, mappel::estimator::ProfileBoundsDataStack::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsDataStack::Ndata, mappel::estimator::ProfileBoundsDataStack::Nparams\_est, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::MLEData::obsI, mappel::estimator::MLEDataStack::obsI, mappel::print\_text\_image(), mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_ub, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, mappel::estimator::MLEDataStack::rllh, mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, mappel::estimator::ProfileBoundsDataStack::target\_rllh\_delta, mappel::estimator::MLEData::theta, and mappel::estimator::MLEDataStack::theta.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_stack().

**9.37.4.21** `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), and mappel::methods::objective::rllh().

**9.37.4.22** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override],[virtual],[inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::estimator::ProfileLikelihoodData::fixed\_idx, mappel::estimator::ProfileLikelihoodData::fixed\_values, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileLikelihoodData::Nfixed, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::ProfileLikelihoodData::Nvalues, mappel::print\_text\_image(), mappel::estimator::ProfileLikelihoodData::profile\_likelihood, mappel::estimator::ProfileLikelihoodData::profile\_parameters, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), and mappel::estimator::Estimator< Model >::record\_walltime().



9.37.4.23 `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_debug_stats ( )`  
`[virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 832 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx`s.

9.37.4.24 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts ( ) const`  
`[inline], [inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.37.4.25 `template<class Model > const Model & mappel::estimator::Estimator< Model >::get_model ( )`  
`[inherited]`

Definition at line 108 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

9.37.4.26 `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_stats ( )`  
`[virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 811 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::Estimator< Model >::num_estimations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`, `mappel::methods::debug::error_bounds_profile_likelihood_debug()`, `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`, `mappel::methods::openmp::error_bounds_profile_likelihood_stack()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`.

**9.37.4.27** `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_backtracks ( ) const`  
`[inline],[inherited]`

Definition at line 441 of file estimator.h.

**9.37.4.28** `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_der_evals ( ) const`  
`[inline],[inherited]`

Definition at line 443 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.37.4.29** `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_fun_evals ( ) const`  
`[inline],[inherited]`

Definition at line 442 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.37.4.30** `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_iterations ( ) const`  
`[inline],[inherited]`

Definition at line 440 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.37.4.31** `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::local_maximize ( const`  
`ModelDataT< Model> & im, StencilT< Model> & stencil, MLEData & data ) [inherited]`

Perform a local maximization to finish off a simulated annealing run.

Definition at line 1145 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

**9.37.4.32** `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::local_maximize ( const`  
`ModelDataT< Model> & im, StencilT< Model> & stencil, MLEDebugData & debug_data ) [inherited]`

Definition at line 1158 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEDebugData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.37.4.33 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize ( const ModelDataT< Model > & im, const IdxVecT & fixed_param_idx, StencilT< Model > & stencil, MLEDebugData & mle ) [inherited]`

Definition at line 1173 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::subroutine::bound_step()`, `mappel::clamp()`, `mappel::estimator::subroutine::compute_bound_scaling_vec()`, `mappel::estimator::subroutine::compute_cauchy_point()`, `mappel::estimator::subroutine::compute_initial_trust_radius()`, `mappel::estimator::subroutine::compute_quadratic_model_value()`, `mappel::estimator::subroutine::compute_scaled_problem()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::fixed_idx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::free_idx`, `mappel::estimator::Estimator< Model >::get_exit_counts()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_iterations()`, `mappel::methods::objective::grad()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::methods::objective::grad2()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters()`, `mappel::methods::objective::hessian()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::is_positive_definite()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::MaxIter`, `mappel::estimator::IterativeMaximizer< Model >::min_profile_bound_residual`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters()`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obs`, `mappel::estimator::MLEDebugData::obs`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s0`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s1`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::estimator::subroutine::solve_TR_subproblem()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::MLEData::theta`, `mappel::estimator::MLEDebugData::theta`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_iterations`, and `mappel::estimator::TrustRegionRadius`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.37.4.34 `template<class Model> virtual void mappel::estimator::IterativeMaximizer< Model >::maximize ( MaximizerData & data ) [protected],[pure virtual],[inherited]`

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.37.4.35 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_backtracks ( )`  
[inherited]

9.37.4.36 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_der_evals ( )`  
[inherited]

9.37.4.37 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_fun_evals ( )`  
[inherited]

9.37.4.38 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_iterations ( )`  
[inherited]

9.37.4.39 `template<class Model > std::string mappel::estimator::NewtonDiagonalMaximizer< Model >::name ( ) const`  
[inline],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 575 of file estimator.h.

9.37.4.40 `template<class Model > bool mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack (`  
`MaximizerData & data, IdxT fixed_idx, double target_rllh, double old_fval, const VecT & fgrad )` [protected],  
[inherited]

Definition at line 943 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_max\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_linear\\_step\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_ratio](#), [mappel::clamp\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_step\\_size\\_ratio](#), [mappel::methods::objective::grad\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im](#), [mappel::estimator::IterativeMaximizer< Model >::max\\_backtracks](#), [mappel::estimator::MaxBacktracks](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_backtrack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_iteration\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore\\_stencil\(\)](#), [mappel::methods::objective::rllh\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save\\_stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved\\_theta\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_stencil\(\)](#), [mappel::square\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step](#), [mappel::estimator::StepSize](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.37.4.41 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code )` [override], [protected], [virtual], [inherited]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::Success`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.37.4.42 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::record_run_statistics ( const MaximizerData & data )` [protected], [inherited]

Definition at line 859 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idxes()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idxes`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nBacktracks`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nIterations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.37.4.43 `template<class Model> void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` [protected], [inherited]

Definition at line 360 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

9.37.4.44 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::solve_profile_bound ( MaximizerData & data, MLEData & mle, double llh_delta, IdxT fixed_idx, IdxT which_bound ) [protected], [virtual], [inherited]`

Definition at line 1137 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::name()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, and `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`.

### 9.37.5 Member Data Documentation

9.37.5.1 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_max_ratio = 0.50 [static], [protected], [inherited]`

Definition at line 460 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.37.5.2 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio = 1e-3 [static], [protected], [inherited]`

Definition at line 461 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.37.5.3 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_ratio = 0.05 [static], [protected], [inherited]`

Definition at line 459 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.37.5.4 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio = 1.0e-9 [static], [protected], [inherited]`

Convergence criteria: tolerance for function-value change.

Definition at line 456 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, and `mappel::estimator::IterativeMaximizer< Model >::get_stats()`.

**9.37.5.5** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio = 1.0e-9` [static], [protected], [inherited]

Convergence criteria: tolerance of relative step size.

Definition at line 457 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.37.5.6** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::DefaultIterations = 100` [static], [inherited]

Definition at line 428 of file estimator.h.

**9.37.5.7** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 299 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.37.5.8** `template<class Model > IdxVecT mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx` [protected], [inherited]

Debugging: Stores last set of `backtrack_idx`s when `data.save_seq==true`.

Definition at line 475 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.37.5.9** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::max_backtracks = 8` [static], [protected], [inherited]

Definition at line 462 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.37.5.10** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::max_iterations` [protected], [inherited]

Definition at line 466 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.37.5.11** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::max_threads`  
`[protected], [inherited]`

Definition at line 324 of file estimator.h.

**9.37.5.12** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model`  
`>::min_eigenvalue_correction_delta = 1e-3 [static], [protected], [inherited]`

Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.

Definition at line 454 of file estimator.h.

**9.37.5.13** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model`  
`>::min_profile_bound_residual = 1e-4 [static], [protected], [inherited]`

Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.

Definition at line 464 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.37.5.14** `template<class Model > const Model& mappel::estimator::Estimator< Model >::model [protected],`  
`[inherited]`

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.37.5.15** `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx [protected],`  
`[inherited]`

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.



**9.37.5.16** `template<class Model> int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected], [inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

**9.37.5.17** `template<class Model> int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.

**9.37.5.18** `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::total_backtracks = 0` [protected], [inherited]

Definition at line 470 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.37.5.19** `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::total_der_evals = 0` [protected], [inherited]

Definition at line 472 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.37.5.20** `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::total_fun_evals = 0` [protected], [inherited]

Definition at line 471 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.37.5.21 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_iterations = 0`  
`[protected], [inherited]`

Definition at line 469 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.37.5.22 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected], [inherited]`

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

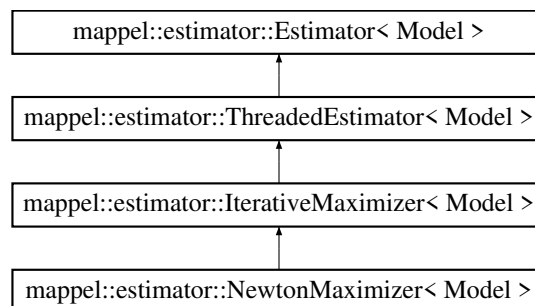
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.38 mappel::estimator::NewtonMaximizer< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::estimator::NewtonMaximizer< Model >`:



### Public Types

- using `MaximizerData` = typename `IterativeMaximizer< Model >::MaximizerData`

## Public Member Functions

- [NewtonMaximizer](#) (const Model &model, int max\_iterations=IterativeMaximizer< Model >::DefaultIterations)
- std::string name () const
- double mean\_iterations ()
- double mean\_backtracks ()
- double mean\_fun\_evals ()
- double mean\_der\_evals ()
- StatsT get\_stats ()
- StatsT get\_debug\_stats ()
- void clear\_stats ()
- int get\_total\_iterations () const
- int get\_total\_backtracks () const
- int get\_total\_fun\_evals () const
- int get\_total\_der\_evals () const
- void local\_maximize (const ModelDataT< Model > &im, StencilT< Model > &stencil, MLEData &data)
  - Perform a local maximization to finish off a simulated annealing run.*
- void local\_maximize (const ModelDataT< Model > &im, StencilT< Model > &stencil, MLEDebugData &debug←\_data)
- void local\_profile\_maximize (const ModelDataT< Model > &im, const IdxVecT &fixed\_param\_idxes, StencilT< Model > &stencil, MLEDebugData &mle)
- void estimate\_max\_stack (const ModelDataStackT< Model > &data, const ParamVecT< Model > &theta\_init←\_stack, MLEDataStack &mle\_data\_stack) override
- void estimate\_profile\_max (const ModelDataT< Model > &data, const ParamVecT< Model > &theta\_init, ProfileLikelihoodData &profile) override
- void estimate\_profile\_bounds\_parallel (const ModelDataT< Model > &data, ProfileBoundsData &bounds\_est) override
- void estimate\_profile\_bounds\_stack (const ModelDataStackT< Model > &data, ProfileBoundsDataStack &bounds\_est\_stack) override
- const Model & get\_model ()
  
- void estimate\_max\_stack (const ModelDataStackT< Model > &data\_stack, MLEDataStack &mle\_data\_stack)
- void estimate\_max (const ModelDataT< Model > &data, const ParamT< Model > &theta\_init, MLEData &mle←\_data, StencilT< Model > &mle\_stencil)
- void estimate\_max (const ModelDataT< Model > &data, const ParamT< Model > &theta\_init, MLEData &mle←\_data)
- void estimate\_max (const ModelDataT< Model > &data, MLEData &mle\_data)
- void estimate\_max\_debug (const ModelDataT< Model > &data, const ParamT< Model > &theta\_init, MLE←\_DebugData &mle\_data, StencilT< Model > &mle\_stencil)
- void estimate\_max\_debug (const ModelDataT< Model > &data, const ParamT< Model > &theta\_init, MLE←\_DebugData &mle\_data)
  
- double estimate\_profile\_max (const ModelDataT< Model > &data, const IdxVecT &fixed\_idxes, const ParamT< Model > &fixed\_theta\_init, StencilT< Model > &theta\_max)
  
- void estimate\_profile\_bounds (const ModelDataT< Model > &data, ProfileBoundsData &bounds\_est)
- void estimate\_profile\_bounds\_debug (const ModelDataT< Model > &data, ProfileBoundsDebugData &bounds←\_est)
  
- IdxVecT get\_exit\_counts () const

## Static Public Attributes

- static const int `DefaultIterations` =100

## Protected Member Functions

- void `record_run_statistics` (const `MaximizerData` &data)
- void `compute_estimate` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil) override
- void `compute_estimate_debug` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil) override
- double `compute_profile_estimate` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, const `IdxVecT` &fixed\_idx, `StencilT`< Model > &theta\_max) override
- void `compute_profile_bound` (const `ModelDataT`< Model > &data, `ProfileBoundsData` &est, const `VecT` &init\_← step, `IdxT` param\_idx, `IdxT` which\_bound) override
- void `compute_profile_bound_debug` (const `ModelDataT`< Model > &data, `ProfileBoundsDebugData` &bounds) override
- bool `backtrack` (`MaximizerData` &data)
- bool `profile_bound_backtrack` (`MaximizerData` &data, `IdxT` fixed\_idx, double target\_rllh, double old\_fval, const `VecT` &fgrad)
- virtual void `maximize` (`MaximizerData` &data)=0
- virtual void `solve_profile_bound` (`MaximizerData` &data, `MLEDData` &mle, double llh\_delta, `IdxT` fixed\_idx, `IdxT` which\_bound)
- bool `convergence_test_grad_ratio` (const `VecT` &grad, double fval)
- bool `convergence_test_step_size` (const `VecT` &new\_theta, const `VecT` &old\_theta)
- void `record_exit_code` (`ExitCode` code) override
- void `record_walltime` (`ClockT`::time\_point start\_walltime, int num\_estimations)

## Protected Attributes

- int `max_iterations`
- int `total_iterations` = 0
- int `total_backtracks` = 0
- int `total_fun_evals` = 0
- int `total_der_evals` = 0
- `IdxVecT` `last_backtrack_idx`

*Debugging: Stores last set of backtrack\_idx when data.save\_seq==true.*

- int `max_threads`
- int `num_threads`
- std::mutex `mtx`
- const Model & `model`
- int `num_estimations` = 0
- double `total_walltime` = 0.
- `IdxVecT` `exit_counts`

## Static Protected Attributes

- static const double `min_eigenvalue_correction_delta` = 1e-3  
*Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.*
- static const double `convergence_min_function_change_ratio` = 1.0e-9  
*Convergence criteria: tolerance for function-value change.*
- static const double `convergence_min_step_size_ratio` = 1.0e-9  
*Convergence criteria: tolerance of relative step size.*
- static const double `backtrack_min_ratio` = 0.05
- static const double `backtrack_max_ratio` = 0.50
- static const double `backtrack_min_linear_step_ratio` = 1e-3
- static const int `max_backtracks` = 8
- static const double `min_profile_bound_residual` = 1e-4  
*Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.*

## 9.38.1 Detailed Description

```
template<class Model>
class mappel::estimator::NewtonMaximizer< Model >
```

Definition at line 582 of file estimator.h.

## 9.38.2 Member Typedef Documentation

9.38.2.1 `template<class Model> using mappel::estimator::NewtonMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 585 of file estimator.h.

## 9.38.3 Constructor &amp; Destructor Documentation

9.38.3.1 `template<class Model> mappel::estimator::NewtonMaximizer< Model >::NewtonMaximizer ( const Model & model, int max_iterations = IterativeMaximizer<Model>::DefaultIterations ) [inline]`

Definition at line 587 of file estimator.h.

## 9.38.4 Member Function Documentation

**9.38.4.1** `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::backtrack ( MaximizerData & data )` `[protected]`, `[inherited]`

Definition at line 870 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio`, `mappel::clamp()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::MaxBacktracks`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::StepSize`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.38.4.2** `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::clear_stats ( )` `[virtual]`, `[inherited]`

Run statistics.

Reimplemented from `mappel::estimator::ThreadedEstimator< Model >`.

Definition at line 848 of file `estimator_impl.h`.

References `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

**9.38.4.3** `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil )` `[override]`, `[protected]`, `[virtual]`, `[inherited]`

Implements `mappel::estimator::Estimator< Model >`.

Definition at line 1043 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

```
9.38.4.4  template<class Model > void mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug ( const
        ModelDataT< Model > & im, const ParamT< Model > & theta_init, MLEDebugData & mle_debug, StencilT<
        Model > & mle_stencil )  [override], [protected], [virtual], [inherited]
```

Virtual estimate\_debug interface

Estimators that produce a sequence of results (e.g. IterativeEstimators) can override this dummy debug implementation.

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1057 of file estimator\_impl.h.

References mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get\_theta\_sequence(), mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get\_theta\_sequence\_rllh(), mappel::estimator::IterativeMaximizer< Model >::maximize(), mappel::estimator::Estimator< Model >::model, mappel::methods::observed\_information(), mappel::estimator::MLEDebugData::obsI, mappel::estimator::IterativeMaximizer< Model >::record\_run\_statistics(), mappel::estimator::MLEDebugData::rllh, mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh, mappel::estimator::MLEDebugData::sequence, mappel::estimator::MLEDebugData::sequence\_rllh, mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil(), mappel::estimator::MLEDebugData::theta, and mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta().

```
9.38.4.5  template<class Model > void mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound ( const
        ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT
        which_bound )  [override], [protected], [virtual], [inherited]
```

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1088 of file estimator\_impl.h.

References mappel::estimator::ProfileBoundsData::estimated\_idx, mappel::estimator::ProfileBoundsData::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::IterativeMaximizer< Model >::record\_run\_statistics(), mappel::estimator::IterativeMaximizer< Model >::solve\_profile\_bound(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, and mappel::estimator::MLEData::theta.

```
9.38.4.6  template<class Model > void mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug (
        const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds )  [override], [protected],
        [virtual], [inherited]
```

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1114 of file estimator\_impl.h.

References mappel::estimator::ProfileBoundsDebugData::estimated\_idx, mappel::estimator::ProfileBoundsDebugData::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsDebugData::Nseq\_lb, mappel::estimator::ProfileBoundsDebugData::Nseq\_ub, mappel::estimator::MLEData::obsI, mappel::estimator::ProfileBoundsDebugData::profile\_lb, mappel::estimator::ProfileBoundsDebugData::profile\_ub, mappel::estimator::IterativeMaximizer< Model >::record\_run\_statistics(), mappel::estimator::ProfileBoundsDebugData::sequence\_lb, mappel::estimator::ProfileBoundsDebugData::sequence\_lb\_rllh, mappel::estimator::ProfileBoundsDebugData::sequence\_ub, mappel::estimator::ProfileBoundsDebugData::sequence\_ub\_rllh, mappel::estimator::IterativeMaximizer< Model >::solve\_profile\_bound(), mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step, mappel::estimator::ProfileBoundsDebugData::target\_rllh\_delta, and mappel::estimator::MLEData::theta.

9.38.4.7 `template<class Model> double mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idx, StencilT< Model > & theta_max ) [override], [protected], [virtual], [inherited]`

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1074 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_fixed\\_parameters\(\)](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

9.38.4.8 `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio ( const VecT & grad, double fval ) [protected], [inherited]`

Definition at line 1015 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_function\\_change\\_ratio](#), [mappel::estimator::GradRatio](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::square\(\)](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.38.4.9 `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size ( const VecT & new_theta, const VecT & old_theta ) [protected], [inherited]`

Definition at line 1027 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_step\\_size\\_ratio](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::estimator::StepSize](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.38.4.10 `template<class Model> void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.



Definition at line 128 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_estimate(), mappel::estimator::Error, mappel::estimator::MLEData::obsI, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, and mappel::estimator::MLEData::theta.

Referenced by mappel::estimator::Estimator< Model >::estimate\_max().

**9.38.4.11** `template<class Model> void mappel::estimator::Estimator< Model >::estimate_max ( const ModelIDataT< Model> & data, const ParamT< Model> & theta_init, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::estimate\_max().

**9.38.4.12** `template<class Model> void mappel::estimator::Estimator< Model >::estimate_max ( const ModelIDataT< Model> & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::estimate\_max(), and mappel::estimator::Estimator< Model >::model.

9.38.4.13 **template<class Model > void mappel::estimator::Estimator< Model >::estimate\_max\_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta\_init, MLEDebugData & mle\_data, StencilT< Model > & mle\_stencil ) [inherited]**

Debug estimation for a single data starting at theta\_init, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The sequence and sequence\_rllh parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including theta\_init and theta\_mle, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.38.4.14 **template<class Model > void mappel::estimator::Estimator< Model >::estimate\_max\_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta\_init, MLEDebugData & mle\_data ) [inherited]**

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_estimate\_debug(), mappel::estimator::Error, mappel::estimator::MLEDebugData::obsI, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_

exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEDebugData::rllh, and mappel::estimator::MLEDebugData::theta.

**9.38.4.15** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::estimate\\_max\\_stack\(\)](#), and [mappel::estimator::Estimator< Model >::model](#).

**9.38.4.16** `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack ) [override],[virtual],[inherited]`

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::MLEDataStack::Ndata](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::MLEData::obsI](#), [mappel::estimator::MLEDataStack::obsI](#),

mappel::print\_text\_image(), mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, mappel::estimator::MLEDataStack::rllh, mappel::estimator::MLEData::theta, and mappel::estimator::MLEDataStack::theta.

**9.38.4.17** `template<class Model> void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_bound(), mappel::estimator::Error, mappel::estimator::ProfileBoundsData::estimated\_idx, mappel::estimator::ProfileBoundsData::initialize\_arrays(), mappel::estimator::ProfileBoundsData::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsData::Nparams\_est, mappel::estimator::MLEData::obs, mappel::print\_text\_image(), mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, and mappel::estimator::MLEData::theta.

Referenced by mappel::methods::error\_bounds\_profile\_likelihood().

**9.38.4.18** `template<class Model> void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_bound\_debug(), mappel::estimator::Error, mappel::estimator::ProfileBoundsDebugData::estimated\_idx, mappel::estimator::ProfileBoundsDebugData::mle, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::ProfileBoundsDebugData::target\_rllh\_delta, and mappel::estimator::MLEData::theta.

Referenced by mappel::methods::debug::error\_bounds\_profile\_likelihood\_debug().

**9.38.4.19** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_bound(), mappel::estimator::Error, mappel::estimator::ProfileBoundsData::estimated\_idx, mappel::estimator::ProfileBoundsData::initialize\_arrays(), mappel::estimator::ProfileBoundsData::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsData::Nparams\_est, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::MLEData::obs, mappel::print\_text\_image(), mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, and mappel::estimator::MLEData::theta.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_parallel().

9.38.4.20 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_bound\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::ProfileBoundsData::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsDataStack::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsData::initialize\\_arrays\(\)](#), [mappel::estimator::ProfileBoundsDataStack::initialize\\_arrays\(\)](#), [mappel::estimator::ProfileBoundsData::mle](#), [mappel::estimator::ProfileBoundsDataStack::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsDataStack::Ndata](#), [mappel::estimator::ProfileBoundsDataStack::Nparams\\_est](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::MLEData::obs](#), [mappel::estimator::MLEDataStack::obs](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ProfileBoundsData::profile\\_lb](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub\\_rllh](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_ub\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_ub](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_ub](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), [mappel::estimator::MLEData::rllh](#), [mappel::estimator::MLEDataStack::rllh](#), [mappel::estimator::subroutine::solve\\_profile\\_initial\\_step\(\)](#), [mappel::estimator::ProfileBoundsData::target\\_rllh\\_delta](#), [mappel::estimator::ProfileBoundsDataStack::target\\_rllh\\_delta](#), [mappel::estimator::MLEData::theta](#), and [mappel::estimator::MLEDataStack::theta](#).

Referenced by [mappel::methods::openmp::error\\_bounds\\_profile\\_likelihood\\_stack\(\)](#).

9.38.4.21 `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::Estimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

9.38.4.22 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override], [virtual], [inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::ProfileLikelihoodData::fixed\\_idx](#)s, [mappel::estimator::ProfileLikelihoodData::fixed\\_values](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileLikelihoodData::Nfixed](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::ProfileLikelihoodData::Nvalues](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ProfileLikelihoodData::profile\\_likelihood](#), [mappel::estimator::ProfileLikelihoodData::profile\\_parameters](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#).

**9.38.4.23** `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_debug_stats ( )`  
`[virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 832 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx`s.

**9.38.4.24** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts ( ) const`  
`[inline], [inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.38.4.25** `template<class Model > const Model & mappel::estimator::Estimator< Model >::get_model ( )`  
`[inherited]`

Definition at line 108 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

**9.38.4.26** `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_stats ( )`  
`[virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 811 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::Estimator< Model >::num_estimations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`, `mappel::methods::debug::error_bounds_profile_likelihood_debug()`, `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`, `mappel::methods::openmp::error_bounds_profile_likelihood_stack()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`.

9.38.4.27 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::get_total_backtracks ( ) const`  
`[inline], [inherited]`

Definition at line 441 of file estimator.h.

9.38.4.28 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals ( ) const`  
`[inline], [inherited]`

Definition at line 443 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.38.4.29 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals ( ) const`  
`[inline], [inherited]`

Definition at line 442 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.38.4.30 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::get_total_iterations ( ) const`  
`[inline], [inherited]`

Definition at line 440 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.38.4.31 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::local_maximize ( const`  
`ModelDataT< Model > & im, StencilT< Model > & stencil, MLEData & data ) [inherited]`

Perform a local maximization to finish off a simulated annealing run.

Definition at line 1145 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.38.4.32 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::local_maximize ( const`  
`ModelDataT< Model > & im, StencilT< Model > & stencil, MLEDebugData & debug_data ) [inherited]`

Definition at line 1158 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::MLEDebugData::theta`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

9.38.4.33 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize ( const ModelDataT< Model > & im, const IdxVecT & fixed_param_idx, StencilT< Model > & stencil, MLEDebugData & mle ) [inherited]`

Definition at line 1173 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::subroutine::bound_step()`, `mappel::clamp()`, `mappel::estimator::subroutine::compute_bound_scaling_vec()`, `mappel::estimator::subroutine::compute_cauchy_point()`, `mappel::estimator::subroutine::compute_initial_trust_radius()`, `mappel::estimator::subroutine::compute_quadratic_model_value()`, `mappel::estimator::subroutine::compute_scaled_problem()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::fixed_idx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::free_idx`, `mappel::estimator::Estimator< Model >::get_exit_counts()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_iterations()`, `mappel::methods::objective::grad()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::methods::objective::grad2()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters()`, `mappel::methods::objective::hessian()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::is_positive_definite()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::MaxIter`, `mappel::estimator::IterativeMaximizer< Model >::min_profile_bound_residual`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters()`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obs`, `mappel::estimator::MLEDebugData::obs`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s0`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s1`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::estimator::subroutine::solve_TR_subproblem()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::MLEData::theta`, `mappel::estimator::MLEDebugData::theta`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_iterations`, and `mappel::estimator::TrustRegionRadius`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.38.4.34 `template<class Model > virtual void mappel::estimator::IterativeMaximizer< Model >::maximize ( MaximizerData & data ) [protected],[pure virtual],[inherited]`

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.



9.38.4.35 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_backtracks ( )`  
[inherited]

9.38.4.36 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_der_evals ( )`  
[inherited]

9.38.4.37 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_fun_evals ( )`  
[inherited]

9.38.4.38 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_iterations ( )`  
[inherited]

9.38.4.39 `template<class Model> std::string mappel::estimator::NewtonMaximizer< Model >::name ( ) const`  
[inline],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 590 of file estimator.h.

9.38.4.40 `template<class Model > bool mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack (`  
`MaximizerData & data, IdxT fixed_idx, double target_rllh, double old_fval, const VecT & fgrad )` [protected],  
[inherited]

Definition at line 943 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_max\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_linear\\_step\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_ratio](#), [mappel::clamp\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_step\\_size\\_ratio](#), [mappel::methods::objective::grad\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im](#), [mappel::estimator::IterativeMaximizer< Model >::max\\_backtracks](#), [mappel::estimator::MaxBacktracks](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_backtrack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_iteration\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore\\_stencil\(\)](#), [mappel::methods::objective::rllh\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save\\_stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved\\_theta\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_stencil\(\)](#), [mappel::square\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step](#), [mappel::estimator::StepSize](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.38.4.41 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code )` [override], [protected], [virtual], [inherited]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::Success`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.38.4.42 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::record_run_statistics ( const MaximizerData & data )` [protected], [inherited]

Definition at line 859 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idxes()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idxes`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nBacktracks`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nIterations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.38.4.43 `template<class Model > void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` [protected], [inherited]

Definition at line 360 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

9.38.4.44 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::solve_profile_bound ( MaximizerData & data, MLEData & mle, double llh_delta, IdxT fixed_idx, IdxT which_bound )` [protected], [virtual], [inherited]

Definition at line 1137 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::name().

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), and mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound\_debug().

### 9.38.5 Member Data Documentation

9.38.5.1 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_max_ratio = 0.50` [static], [protected], [inherited]

Definition at line 460 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.38.5.2 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio = 1e-3` [static], [protected], [inherited]

Definition at line 461 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::backtrack(), and mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.38.5.3 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_ratio = 0.05` [static], [protected], [inherited]

Definition at line 459 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.38.5.4 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio = 1.0e-9` [static], [protected], [inherited]

Convergence criteria: tolerance for function-value change.

Definition at line 456 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::convergence\_test\_grad\_ratio(), and mappel::estimator::IterativeMaximizer< Model >::get\_stats().

**9.38.5.5** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio = 1.0e-9` [static], [protected], [inherited]

Convergence criteria: tolerance of relative step size.

Definition at line 457 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.38.5.6** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::DefaultIterations = 100` [static], [inherited]

Definition at line 428 of file estimator.h.

**9.38.5.7** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 299 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.38.5.8** `template<class Model > IdxVecT mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx` [protected], [inherited]

Debugging: Stores last set of `backtrack_idx`s when `data.save_seq==true`.

Definition at line 475 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.38.5.9** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::max_backtracks = 8` [static], [protected], [inherited]

Definition at line 462 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.38.5.10** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::max_iterations` [protected], [inherited]

Definition at line 466 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.38.5.11 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::max_threads`  
`[protected], [inherited]`

Definition at line 324 of file estimator.h.

9.38.5.12 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model`  
`>::min_eigenvalue_correction_delta = 1e-3 [static], [protected], [inherited]`

Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.

Definition at line 454 of file estimator.h.

9.38.5.13 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model`  
`>::min_profile_bound_residual = 1e-4 [static], [protected], [inherited]`

Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.

Definition at line 464 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.38.5.14 `template<class Model > const Model& mappel::estimator::Estimator< Model >::model` `[protected],`  
`[inherited]`

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

9.38.5.15 `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` `[protected],`  
`[inherited]`

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.38.5.16** `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected], [inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

**9.38.5.17** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.

**9.38.5.18** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_backtracks = 0` [protected], [inherited]

Definition at line 470 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.38.5.19** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_der_evals = 0` [protected], [inherited]

Definition at line 472 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.38.5.20** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_fun_evals = 0` [protected], [inherited]

Definition at line 471 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.38.5.21 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_iterations = 0`  
`[protected], [inherited]`

Definition at line 469 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.38.5.22 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected], [inherited]`

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

The documentation for this class was generated from the following files:

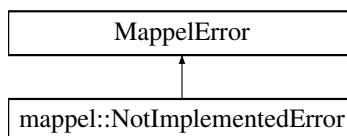
- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.39 mappel::NotImplementedError Struct Reference

Feature not yet implemented.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/util.h>
```

Inheritance diagram for `mappel::NotImplementedError`:



### Public Member Functions

- [NotImplementedError](#) (std::string message)

### 9.39.1 Detailed Description

Feature not yet implemented.

Definition at line 111 of file util.h.

### 9.39.2 Constructor & Destructor Documentation

#### 9.39.2.1 mappel::NotImplementedError::NotImplementedError ( std::string *message* ) [inline]

Definition at line 113 of file util.h.

The documentation for this struct was generated from the following file:

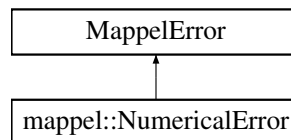
- [util.h](#)

## 9.40 mappel::NumericalError Struct Reference

Expected numerical condition does not hold.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/util.h>
```

Inheritance diagram for mappel::NumericalError:



### Public Member Functions

- [NumericalError](#) (std::string message)

#### 9.40.1 Detailed Description

Expected numerical condition does not hold.

Definition at line 97 of file util.h.

### 9.40.2 Constructor & Destructor Documentation

#### 9.40.2.1 mappel::NumericalError::NumericalError ( std::string *message* ) [inline]

Definition at line 99 of file util.h.

The documentation for this struct was generated from the following file:

- [util.h](#)



## 9.41 `omp_exception_catcher::impl_::OMPEXceptionCatcher<_dummy>` Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/OMPEXceptionCatcher/OMPEXceptionCatcher.h>
```

### Public Member Functions

- [OMPEXceptionCatcher](#) ()
- [OMPEXceptionCatcher](#) ([Strategy](#) strategy\_)
- void [rethrow](#) () const
- template<class Function, class... Parameters>  
void [run](#) (Function func, Parameters...params)

### Static Public Member Functions

- static void [setGlobalDefaultStrategy](#) ([Strategy](#) s)

#### 9.41.1 Detailed Description

```
template<class _dummy = void>  
class omp_exception_catcher::impl_::OMPEXceptionCatcher<_dummy>
```

Implementation of [OMPEXceptionCatcher](#)

Note: The template variable is a dummy. It exists solely to allow this class to be a template, which makes it header-only and allows static member initialization to be defined in the header file.

Definition at line 47 of file `OMPEXceptionCatcher.h`.

#### 9.41.2 Constructor & Destructor Documentation

9.41.2.1 `template<class _dummy = void> omp_exception_catcher::impl_::OMPEXceptionCatcher<_dummy>::OMPEXceptionCatcher ( ) [inline]`

Construct a new [OMPEXceptionCatcher](#) using the `GlobalDefaultStrategy`

Definition at line 55 of file `OMPEXceptionCatcher.h`.

9.41.2.2 `template<class _dummy = void> omp_exception_catcher::impl_::OMPEXceptionCatcher<_dummy>::OMPEXceptionCatcher ( Strategy strategy_ ) [inline]`

Construct a new [OMPEXceptionCatcher](#) using the given strategy

Definition at line 59 of file `OMPEXceptionCatcher.h`.

### 9.41.3 Member Function Documentation

**9.41.3.1** `template<class _dummy = void> void omp_exception_catcher::impl_::OMPEXceptionCatcher< _dummy >::rethrow ( ) const [inline]`

Rethrow any stored exceptions Should only be called from single-threaded blocks of code

Definition at line 64 of file OMPEXceptionCatcher.h.

References `omp_exception_catcher::RethrowFirst`.

Referenced by `mappel::methods::openmp::cr_lower_bound_stack()`, `mappel::methods::openmp::error_bounds_expected_stack()`, `mappel::methods::openmp::estimate_posterior_stack()`, `mappel::methods::openmp::expected_information_stack()`, `mappel::methods::objective::openmp::grad_stack()`, `mappel::methods::objective::openmp::hessian_stack()`, `mappel::methods::objective::openmp::llh_stack()`, `mappel::methods::openmp::model_image_stack()`, `mappel::methods::objective::openmp::negative_definite_hessian_stack()`, `mappel::methods::objective::openmp::rllh_stack()`, `mappel::methods::openmp::sample_prior_stack()`, and `mappel::methods::openmp::simulate_image_stack()`.

**9.41.3.2** `template<class _dummy = void> template<class Function , class... Parameters> void omp_exception_catcher::impl_::OMPEXceptionCatcher< _dummy >::run ( Function func, Parameters... params ) [inline]`

Run a function in parallel code and prevent exceptions escaping.

Runs any function with any set of parameters and applies the chosen exception catching Strategy to prevent any exceptions escaping. This function is thread-safe designed to be called in parallel code blocks.

#### Parameters

in	<i>func</i>	function to call
in	<i>params...</i>	Possibly empty variadic set of parameters to call.

Definition at line 76 of file OMPEXceptionCatcher.h.

References `omp_exception_catcher::Abort`, `omp_exception_catcher::Continue`, `omp_exception_catcher::DoNotTry`, and `omp_exception_catcher::RethrowFirst`.

Referenced by `mappel::methods::openmp::cr_lower_bound_stack()`, `mappel::methods::openmp::error_bounds_expected_stack()`, `mappel::methods::openmp::error_bounds_observed_stack()`, `mappel::methods::openmp::estimate_posterior_stack()`, `mappel::methods::openmp::expected_information_stack()`, `mappel::methods::objective::openmp::grad_stack()`, `mappel::methods::objective::openmp::hessian_stack()`, `mappel::methods::objective::openmp::llh_stack()`, `mappel::methods::openmp::model_image_stack()`, `mappel::methods::objective::openmp::negative_definite_hessian_stack()`, `mappel::methods::objective::openmp::rllh_stack()`, `mappel::methods::openmp::sample_prior_stack()`, and `mappel::methods::openmp::simulate_image_stack()`.

**9.41.3.3** `template<class _dummy = void> static void omp_exception_catcher::impl_::OMPEXceptionCatcher< _dummy >::setGlobalDefaultStrategy ( Strategy s ) [inline], [static]`

Definition at line 51 of file OMPEXceptionCatcher.h.

The documentation for this class was generated from the following file:

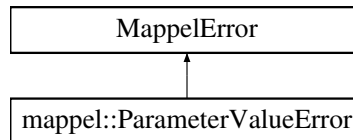
- [OMPEXceptionCatcher.h](#)

## 9.42 mappel::ParameterValueError Struct Reference

Parameter value is not valid.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/util.h>
```

Inheritance diagram for mappel::ParameterValueError:



### Public Member Functions

- [ParameterValueError](#) (std::string message)

#### 9.42.1 Detailed Description

Parameter value is not valid.

Definition at line 69 of file util.h.

#### 9.42.2 Constructor & Destructor Documentation

##### 9.42.2.1 mappel::ParameterValueError::ParameterValueError ( std::string *message* ) `[inline]`

Definition at line 71 of file util.h.

The documentation for this struct was generated from the following file:

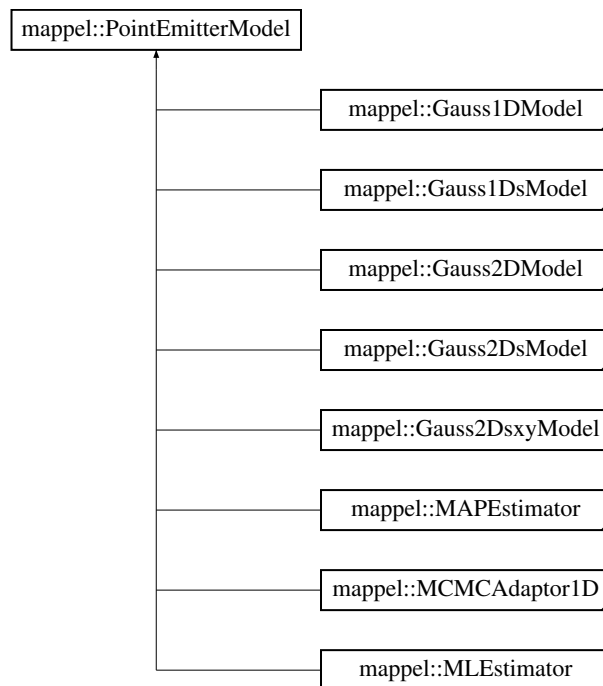
- [util.h](#)

### 9.43 mappel::PointEmitterModel Class Reference

A virtual Base type for point emitter localization models.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/PointEmitterModel.h>
```

Inheritance diagram for mappel::PointEmitterModel:



#### Public Types

- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat

#### Public Member Functions

- [StatsT get\\_stats](#) () const
- [IdxT get\\_num\\_params](#) () const
- void [check\\_param\\_shape](#) (const [ParamT](#) &theta) const
- void [check\\_param\\_shape](#) (const [ParamVecT](#) &theta) const
- void [check\\_psf\\_sigma](#) (double psf\_sigma) const
- void [check\\_psf\\_sigma](#) (const [VecT](#) &psf\_sigma) const
- [ParamT make\\_param](#) () const
- [ParamVecT make\\_param\\_stack](#) ([IdxT](#) n) const
- [MatT make\\_param\\_mat](#) () const
- [CubeT make\\_param\\_mat\\_stack](#) ([IdxT](#) n) const

- template<class FillT >  
  ParamT make\_param (FillT fill) const
- template<class FillT >  
  ParamVecT make\_param\_stack (IdxT n, FillT fill) const
- template<class FillT >  
  MatT make\_param\_mat (FillT fill) const
- template<class FillT >  
  CubeT make\_param\_mat\_stack (IdxT n, FillT fill) const
- CompositeDist & get\_prior ()
- const CompositeDist & get\_prior () const
- void set\_prior (CompositeDist &&prior\_)
- void set\_prior (const CompositeDist &prior\_)
- IdxT get\_num\_hyperparams () const
- void set\_hyperparams (const VecT &hyperparams)
- VecT get\_hyperparams () const
- bool has\_hyperparam (const std::string &name) const
- double get\_hyperparam\_value (const std::string &name) const
- int get\_hyperparam\_index (const std::string &name) const
- void set\_hyperparam\_value (const std::string &name, double value)
- void rename\_hyperparam (const std::string &old\_name, const std::string &new\_name)
- StringVecT get\_param\_names () const
- void set\_param\_names (const StringVecT &desc)
- StringVecT get\_hyperparam\_names () const
- void set\_hyperparam\_names (const StringVecT &desc)
- template<class RngT >  
  ParamT sample\_prior (RngT &rng) const
- ParamT sample\_prior () const
- void set\_bounds (const ParamT &lbound, const ParamT &ubound)
- void set\_lbound (const ParamT &lbound)
- void set\_ubound (const ParamT &ubound)
- const ParamT & get\_lbound () const
- const ParamT & get\_ubound () const
- bool theta\_in\_bounds (const ParamT &theta) const
- void bound\_theta (ParamT &theta, double epsilon=bounds\_epsilon) const
- ParamT bounded\_theta (const ParamT &theta, double epsilon=bounds\_epsilon) const
- ParamT reflected\_theta (const ParamT &theta) const
- BoolVecT theta\_stack\_in\_bounds (const ParamVecT &theta) const
- ParamVecT bounded\_theta\_stack (const ParamVecT &theta, double epsilon=bounds\_epsilon) const
- ParamVecT reflected\_theta\_stack (const ParamVecT &theta) const

#### Static Public Member Functions

- static prior\_hessian::TruncatedNormalDist make\_prior\_component\_position\_normal (IdxT size, double pos\_↵, double sigma=default\_sigma\_pos)
- static prior\_hessian::ScaledSymmetricBetaDist make\_prior\_component\_position\_beta (IdxT size, double pos\_↵, double beta=default\_beta\_pos)
- static prior\_hessian::TruncatedGammaDist make\_prior\_component\_intensity (double mean=default\_mean\_↵, double kappa=default\_intensity\_kappa)
- static prior\_hessian::TruncatedParetoDist make\_prior\_component\_sigma (double min\_sigma, double max\_↵, double sigma, double alpha=default\_alpha\_sigma)
- static void set\_rng\_seed (RngSeedT seed)
- static ParallelRngManagerT & get\_rng\_manager ()
- static ParallelRngGeneratorT & get\_rng\_generator ()

### Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds\\_epsilon](#) = 1.0E-6
- static const double [global\\_min\\_psf\\_sigma](#) = 1E-1
- static const double [global\\_max\\_psf\\_sigma](#) = 1E2
- static const double [default\\_beta\\_pos](#) = 3
- static const double [default\\_sigma\\_pos](#) = 1
- static const double [default\\_mean\\_l](#) = 300
- static const double [default\\_max\\_l](#) = INFINITY
- static const double [default\\_intensity\\_kappa](#) = 2
- static const double [default\\_pixel\\_mean\\_bg](#) = 4
- static const double [default\\_alpha\\_sigma](#) = 2

### Protected Member Functions

- [PointEmitterModel](#) ()
- [PointEmitterModel](#) (const CompositeDist &prior\_)
- [PointEmitterModel](#) (CompositeDist &&prior\_)
- [PointEmitterModel](#) (const [PointEmitterModel](#) &)
- [PointEmitterModel](#) ([PointEmitterModel](#) &&)
- [PointEmitterModel](#) & operator= (const [PointEmitterModel](#) &)
- [PointEmitterModel](#) & operator= ([PointEmitterModel](#) &&)

### Protected Attributes

- CompositeDist [prior](#)
- [IdxT](#) [num\\_params](#)
- [IdxT](#) [num\\_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)

#### 9.43.1 Detailed Description

A virtual Base type for point emitter localization models.

<Composite distribution from `prior_hessian::` for representing priorsInitialized with a prior as a `PriorHessian::`← CompositeDist object, this sets the dimensionality (`num_params`) and `num_hyperparams`, and the associated descriptions.

Box-type bounding constraints are controlled with the [set\\_bounds\(\)](#) method.

Of note some of the common MCMC variables are rooted here in the inheritance tree.

Definition at line 44 of file `PointEmitterModel.h`.

## 9.43.2 Member Typedef Documentation

## 9.43.2.1 using mappel::PointEmitterModel::ParamT = arma::vec

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

## 9.43.2.2 using mappel::PointEmitterModel::ParamVecT = arma::mat

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

## 9.43.3 Constructor &amp; Destructor Documentation

## 9.43.3.1 mappel::PointEmitterModel::PointEmitterModel ( ) [protected]

Definition at line 31 of file PointEmitterModel.cpp.

## 9.43.3.2 mappel::PointEmitterModel::PointEmitterModel ( const CompositeDist &amp; prior\_ ) [explicit], [protected]

Definition at line 43 of file PointEmitterModel.cpp.

## 9.43.3.3 mappel::PointEmitterModel::PointEmitterModel ( CompositeDist &amp;&amp; prior\_ ) [explicit], [protected]

Definition at line 37 of file PointEmitterModel.cpp.

## 9.43.3.4 mappel::PointEmitterModel::PointEmitterModel ( const PointEmitterModel &amp; o ) [protected]

Definition at line 49 of file PointEmitterModel.cpp.

References prior.

## 9.43.3.5 mappel::PointEmitterModel::PointEmitterModel ( PointEmitterModel &amp;&amp; o ) [protected]

Definition at line 55 of file PointEmitterModel.cpp.

## 9.43.4 Member Function Documentation

## 9.43.4.1 void mappel::PointEmitterModel::bound\_theta ( ParamT &amp; theta, double epsilon = bounds\_epsilon ) const

Definition at line 248 of file PointEmitterModel.cpp.

References check\_param\_shape(), lbound, num\_params, and ubound.

**9.43.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded\_theta ( const ParamT & *theta*, double *epsilon* = bounds\_epsilon ) const**

Definition at line 265 of file PointEmitterModel.cpp.

References check\_param\_shape(), lbound, num\_params, and ubound.

Referenced by bounded\_theta\_stack().

**9.43.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded\_theta\_stack ( const ParamVecT & *theta*, double *epsilon* = bounds\_epsilon ) const**

Definition at line 306 of file PointEmitterModel.cpp.

References bounded\_theta(), check\_param\_shape(), and make\_param\_stack().

**9.43.4.4 void mappel::PointEmitterModel::check\_param\_shape ( const ParamT & *theta* ) const**

Definition at line 167 of file PointEmitterModel.cpp.

References num\_params.

Referenced by bound\_theta(), bounded\_theta(), bounded\_theta\_stack(), reflected\_theta(), reflected\_theta\_stack(), theta\_in\_bounds(), and theta\_stack\_in\_bounds().

**9.43.4.5 void mappel::PointEmitterModel::check\_param\_shape ( const ParamVecT & *theta* ) const**

Definition at line 176 of file PointEmitterModel.cpp.

References num\_params.

**9.43.4.6 void mappel::PointEmitterModel::check\_psf\_sigma ( double *psf\_sigma* ) const**

Definition at line 185 of file PointEmitterModel.cpp.

References global\_max\_psf\_sigma, and global\_min\_psf\_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set\_max\_sigma(), mappel::Gauss2DsModel::set\_max\_sigma(), mappel::Gauss1DsModel::set\_min\_sigma(), mappel::Gauss2DsModel::set\_min\_sigma(), mappel::Gauss1DModel::set\_psf\_sigma(), and mappel::Gauss2DModel::set\_psf\_sigma().

**9.43.4.7 void mappel::PointEmitterModel::check\_psf\_sigma ( const VecT & *psf\_sigma* ) const**

Definition at line 197 of file PointEmitterModel.cpp.

References global\_max\_psf\_sigma, and global\_min\_psf\_sigma.



9.43.4.8 `int mappel::PointEmitterModel::get_hyperparam_index ( const std::string & name ) const [inline]`

Definition at line 236 of file PointEmitterModel.h.

References prior.

9.43.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names ( ) const [inline]`

Definition at line 256 of file PointEmitterModel.h.

References prior.

9.43.4.10 `double mappel::PointEmitterModel::get_hyperparam_value ( const std::string & name ) const [inline]`

Definition at line 232 of file PointEmitterModel.h.

References prior.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.43.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams ( ) const [inline]`

Definition at line 224 of file PointEmitterModel.h.

References prior.

9.43.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound ( ) const [inline]`

Definition at line 212 of file PointEmitterModel.h.

References lbound.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.43.4.13 `IdxT mappel::PointEmitterModel::get_num_hyperparams ( ) const [inline]`

Definition at line 208 of file PointEmitterModel.h.

References num\_hyperparams.

9.43.4.14 `IdxT mappel::PointEmitterModel::get_num_params ( ) const [inline]`

Definition at line 160 of file PointEmitterModel.h.

References num\_params.

#### 9.43.4.15 `StringVecT mappel::PointEmitterModel::get_param_names ( ) const [inline]`

Definition at line 248 of file `PointEmitterModel.h`.

References prior.

#### 9.43.4.16 `CompositeDist & mappel::PointEmitterModel::get_prior ( ) [inline]`

Definition at line 200 of file `PointEmitterModel.h`.

References prior.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

#### 9.43.4.17 `const CompositeDist & mappel::PointEmitterModel::get_prior ( ) const [inline]`

Definition at line 204 of file `PointEmitterModel.h`.

References prior.

#### 9.43.4.18 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ( ) [static]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

#### 9.43.4.19 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ( ) [static]`

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

#### 9.43.4.20 `StatsT mappel::PointEmitterModel::get_stats ( ) const`

Definition at line 125 of file `PointEmitterModel.cpp`.

References `lbound`, `num_hyperparams`, `num_params`, `prior`, `mappel::rng_manager`, and `ubound`.

Referenced by `mappel::Gauss1DModel::get_stats()`, `mappel::Gauss1DsModel::get_stats()`, `mappel::Gauss2DModel::get_stats()`, and `mappel::Gauss2DsModel::get_stats()`.

#### 9.43.4.21 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound ( ) const [inline]`

Definition at line 216 of file `PointEmitterModel.h`.

References `ubound`.

Referenced by `mappel::Gauss2DsxyModel::get_max_sigma_ratio()`, `mappel::Gauss2DsModel::get_max_sigma_ratio()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

9.43.4.22 `bool mappel::PointEmitterModel::has_hyperparam ( const std::string & name ) const [inline]`

Definition at line 228 of file PointEmitterModel.h.

References prior.

9.43.4.23 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( ) const [inline]`

Definition at line 164 of file PointEmitterModel.h.

References num\_params.

Referenced by mappel::Gauss1DModel::initial\_theta\_estimate(), mappel::Gauss1DsModel::initial\_theta\_estimate(), mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsxyModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::pixel\_hess\_update(), mappel::Gauss1DsModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_hess\_update(), and mappel::Gauss2DsModel::pixel\_hess\_update().

9.43.4.24 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ( FillT fill ) const`

Definition at line 181 of file PointEmitterModel.h.

References num\_params.

9.43.4.25 `MatT mappel::PointEmitterModel::make_param_mat ( ) const [inline]`

Definition at line 172 of file PointEmitterModel.h.

References num\_params.

9.43.4.26 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat ( FillT fill ) const`

Definition at line 191 of file PointEmitterModel.h.

References num\_params.

9.43.4.27 `CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n ) const [inline]`

Definition at line 176 of file PointEmitterModel.h.

References num\_params.

9.43.4.28 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack ( IdxT n, FillT fill ) const`

Definition at line 196 of file PointEmitterModel.h.

References num\_params.

#### 9.43.4.29 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n ) const` `[inline]`

Definition at line 168 of file `PointEmitterModel.h`.

References `num_params`.

Referenced by `bounded_theta_stack()`, and `reflected_theta_stack()`.

#### 9.43.4.30 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack ( IdxT n, FillT fill ) const`

Definition at line 186 of file `PointEmitterModel.h`.

References `num_params`.

#### 9.43.4.31 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity ( double mean = default_mean_I, double kappa = default_intensity_kappa )` `[static]`

Definition at line 98 of file `PointEmitterModel.cpp`.

References `default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

#### 9.43.4.32 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta ( IdxT size, double pos_beta = default_beta_pos )` `[static]`

Definition at line 92 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

#### 9.43.4.33 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal ( IdxT size, double pos_sigma = default_sigma_pos )` `[static]`

Definition at line 85 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.43.4.34 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma ( double min_sigma, double max_sigma, double alpha = default_alpha_sigma ) [static]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

9.43.4.35 `PointEmitterModel & mappel::PointEmitterModel::operator= ( const PointEmitterModel & o ) [protected]`

Definition at line 61 of file PointEmitterModel.cpp.

References `prior`.

Referenced by `mappel::Gauss1DMAP::operator=()`, `mappel::Gauss1DMLE::operator=()`, `mappel::Gauss1DsMLE::operator=()`, `mappel::Gauss1DsMAP::operator=()`, `mappel::Gauss2DsMAP::operator=()`, `mappel::Gauss2DsMLE::operator=()`, `mappel::Gauss2DMLE::operator=()`, and `mappel::Gauss2DMAP::operator=()`.

9.43.4.36 `PointEmitterModel & mappel::PointEmitterModel::operator= ( PointEmitterModel && o ) [protected]`

Definition at line 68 of file PointEmitterModel.cpp.

References `lbound`, `num_hyperparams`, `num_params`, `prior`, and `ubound`.

9.43.4.37 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta ( const ParamT & theta ) const`

Definition at line 276 of file PointEmitterModel.cpp.

References `check_param_shape()`, `lbound`, `num_params`, and `ubound`.

Referenced by `reflected_theta_stack()`.

9.43.4.38 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack ( const ParamVecT & theta ) const`

Definition at line 316 of file PointEmitterModel.cpp.

References `check_param_shape()`, `make_param_stack()`, and `reflected_theta()`.

9.43.4.39 `void mappel::PointEmitterModel::rename_hyperparam ( const std::string & old_name, const std::string & new_name ) [inline]`

Definition at line 244 of file PointEmitterModel.h.

References `prior`.

9.43.4.40 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( RngT & rng ) const`

Definition at line 264 of file PointEmitterModel.h.

References prior.

9.43.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ( ) const [inline]`

Definition at line 268 of file PointEmitterModel.h.

References prior, and mappel::rng\_manager.

9.43.4.42 `void mappel::PointEmitterModel::set_bounds ( const ParamT & lbound_, const ParamT & ubound_ )`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References bounds\_epsilon, lbound, num\_params, prior, and ubound.

9.43.4.43 `void mappel::PointEmitterModel::set_hyperparam_names ( const StringVecT & desc ) [inline]`

Definition at line 260 of file PointEmitterModel.h.

References prior.

9.43.4.44 `void mappel::PointEmitterModel::set_hyperparam_value ( const std::string & name, double value ) [inline]`

Definition at line 240 of file PointEmitterModel.h.

References prior.

9.43.4.45 `void mappel::PointEmitterModel::set_hyperparams ( const VecT & hyperparams ) [inline]`

Definition at line 220 of file PointEmitterModel.h.

References prior.

Referenced by mappel::Gauss2DModel::set\_hyperparams(), and mappel::Gauss2DsModel::set\_hyperparams().

9.43.4.46 `void mappel::PointEmitterModel::set_lbound ( const ParamT & lbound )`

Definition at line 226 of file PointEmitterModel.cpp.

References bounds\_epsilon, lbound, num\_params, prior, and ubound.

Referenced by mappel::Gauss1DsModel::set\_min\_sigma().

9.43.4.47 `void mappel::PointEmitterModel::set_param_names ( const StringVecT & desc ) [inline]`

Definition at line 252 of file PointEmitterModel.h.

References prior.

9.43.4.48 `void mappel::PointEmitterModel::set_prior ( CompositeDist && prior_ )`

Definition at line 158 of file PointEmitterModel.cpp.

References lbound, num\_hyperparams, num\_params, prior, and ubound.

Referenced by mappel::Gauss2DModel::set\_prior(), and mappel::Gauss2DsModel::set\_prior().

9.43.4.49 `void mappel::PointEmitterModel::set_prior ( const CompositeDist & prior_ )`

Definition at line 149 of file PointEmitterModel.cpp.

References lbound, num\_hyperparams, num\_params, prior, and ubound.

9.43.4.50 `void mappel::PointEmitterModel::set_rng_seed ( RngSeedT seed ) [static]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng\_manager.

9.43.4.51 `void mappel::PointEmitterModel::set_ubound ( const ParamT & ubound )`

Definition at line 237 of file PointEmitterModel.cpp.

References bounds\_epsilon, lbound, num\_params, prior, and ubound.

Referenced by mappel::Gauss1DsModel::set\_max\_sigma(), and mappel::Gauss2DsModel::set\_max\_sigma\_ratio().

9.43.4.52 `bool mappel::PointEmitterModel::theta_in_bounds ( const ParamT & theta ) const`

Definition at line 257 of file PointEmitterModel.cpp.

References check\_param\_shape(), lbound, num\_params, and ubound.

Referenced by mappel::Gauss2DModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss1DModel::make\_stencil(), mappel::Gauss1DsModel::make\_stencil(), mappel::Gauss2DModel::make\_stencil(), mappel::Gauss2DsxyModel::make\_stencil(), mappel::Gauss2DsModel::make\_stencil(), and theta\_stack\_in\_bounds().

9.43.4.53 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds ( const ParamVecT & theta ) const`

Definition at line 296 of file PointEmitterModel.cpp.

References check\_param\_shape(), and theta\_in\_bounds().

### 9.43.5 Member Data Documentation

**9.43.5.1** `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file `PointEmitterModel.h`.

Referenced by `set_bounds()`, `set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `set_ubound()`.

**9.43.5.2** `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file `PointEmitterModel.h`.

**9.43.5.3** `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file `PointEmitterModel.h`.

**9.43.5.4** `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static]`

Default shape for intensity gamma distributions

Definition at line 60 of file `PointEmitterModel.h`.

**9.43.5.5** `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static]`

Default maximum emitter intensity

Definition at line 59 of file `PointEmitterModel.h`.

Referenced by `make_prior_component_intensity()`.

**9.43.5.6** `const double mappel::PointEmitterModel::default_mean_l = 300` `[static]`

Default emitter intensity mean

Definition at line 58 of file `PointEmitterModel.h`.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.



9.43.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.43.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

9.43.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static]`

Estimator name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.43.5.10 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `check_psf_sigma()`.

9.43.5.11 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `check_psf_sigma()`.

9.43.5.12 `ParamT mappel::PointEmitterModel::lbound` `[protected]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by `bound_theta()`, `bounded_theta()`, `get_lbound()`, `get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `operator=()`, `reflected_theta()`, `set_bounds()`, `set_lbound()`, `set_prior()`, `set_ubound()`, and `theta_in_bounds()`.

#### 9.43.5.13 IdxT mappel::PointEmitterModel::num\_hyperparams [protected]

Definition at line 147 of file PointEmitterModel.h.

Referenced by `get_num_hyperparams()`, `get_stats()`, `operator=()`, and `set_prior()`.

#### 9.43.5.14 IdxT mappel::PointEmitterModel::num\_params [protected]

Definition at line 146 of file PointEmitterModel.h.

Referenced by `bound_theta()`, `bounded_theta()`, `check_param_shape()`, `get_num_params()`, `get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `make_param()`, `make_param_mat()`, `make_param_mat_stack()`, `make_param_stack()`, `operator=()`, `reflected_theta()`, `set_bounds()`, `set_lbound()`, `set_prior()`, `set_ubound()`, and `theta_in_bounds()`.

#### 9.43.5.15 CompositeDist mappel::PointEmitterModel::prior [protected]

Definition at line 145 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `get_hyperparam_index()`, `get_hyperparam_names()`, `get_hyperparam_value()`, `get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `get_param_names()`, `get_prior()`, `get_stats()`, `has_hyperparam()`, `operator=()`, `PointEmitterModel()`, `rename_hyperparam()`, `sample_prior()`, `set_bounds()`, `set_hyperparam_names()`, `set_hyperparam_value()`, `set_hyperparams()`, `set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `set_param_names()`, `set_prior()`, and `set_ubound()`.

#### 9.43.5.16 ParamT mappel::PointEmitterModel::ubound [protected]

Definition at line 148 of file PointEmitterModel.h.

Referenced by `bound_theta()`, `bounded_theta()`, `get_stats()`, `get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `operator=()`, `reflected_theta()`, `set_bounds()`, `set_lbound()`, `set_prior()`, `set_ubound()`, and `theta_in_bounds()`.

The documentation for this class was generated from the following files:

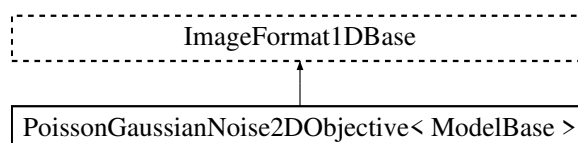
- [PointEmitterModel.h](#)
- [PointEmitterModel.cpp](#)

## 9.44 PoissonGaussianNoise2DObjective< ModelBase > Class Template Reference

A Base type for point emitter localization models that use 2d images.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/PoissonGaussianNoise2DObjective.h>
```

Inheritance diagram for `PoissonGaussianNoise2DObjective< ModelBase >`:



## Public Types

- using [CoordIdxT](#) = uint32\_t
- using [CoordT](#) = arma::vec< uint32\_t >
- using [CoordStackT](#) = arma::mat< uint32\_t >
- using [ModelDataT](#) = std::pair< ImageT, [CoordT](#) >
- using [ModelDataStackT](#) = std::pair< ImageStackT, [CoordStackT](#) >

## Public Member Functions

- [PoissonGaussianNoise2DObjective](#) (const ImageSizeVecT &size, const ImageT &[sensor\\_gain\\_map](#), const ImageT &[sensor\\_bg\\_map](#))

## Public Attributes

- ImageT [sensor\\_gain\\_map](#)
- ImageT [sensor\\_bg\\_map](#)

## Static Public Attributes

- static const StringVecT [estimator\\_names](#)

## 9.44.1 Detailed Description

```
template<typename ModelBase>
class PoissonGaussianNoise2DObjective< ModelBase >
```

A Base type for point emitter localization models that use 2d images.

We don't assume much here, so that it is possible to have a wide range of 2D models

Definition at line 23 of file PoissonGaussianNoise2DObjective.h.

## 9.44.2 Member Typedef Documentation

9.44.2.1 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordIdxT = uint32_t`

Definition at line 26 of file PoissonGaussianNoise2DObjective.h.

9.44.2.2 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordStackT = arma::mat<uint32_t>`

Definition at line 28 of file PoissonGaussianNoise2DObjective.h.

9.44.2.3 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordT = arma::vec<uint32_t>`

Definition at line 27 of file PoissonGaussianNoise2DObjective.h.

9.44.2.4 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::ModelDataStackT = std::pair<ImageStackT,CoordStackT>`

Objective function data stack type: 2D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 30 of file PoissonGaussianNoise2DObjective.h.

9.44.2.5 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::ModelDataT = std::pair<ImageT, CoordT>`

Objective function data type: 2D double precision image, gain-corrected to approximate photons counts

Definition at line 29 of file PoissonGaussianNoise2DObjective.h.

#### 9.44.3 Constructor & Destructor Documentation

9.44.3.1 `template<typename ModelBase > PoissonGaussianNoise2DObjective< ModelBase >::PoissonGaussianNoise2DObjective ( const ImageSizeVecT & size, const ImageT & sensor_gain_map, const ImageT & sensor_bg_map )`

#### 9.44.4 Member Data Documentation

9.44.4.1 `template<typename ModelBase > const std::vector< std::string > PoissonGaussianNoise2DObjective< ModelBase >::estimator_names [static]`

Definition at line 25 of file PoissonGaussianNoise2DObjective.h.

9.44.4.2 `template<typename ModelBase > ImageT PoissonGaussianNoise2DObjective< ModelBase >::sensor_bg_map`

Definition at line 34 of file PoissonGaussianNoise2DObjective.h.

9.44.4.3 `template<typename ModelBase > ImageT PoissonGaussianNoise2DObjective< ModelBase >::sensor_gain_map`

Definition at line 33 of file PoissonGaussianNoise2DObjective.h.

The documentation for this class was generated from the following files:

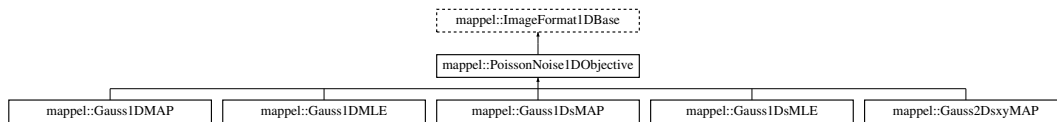
- [PoissonGaussianNoise2DObjective.h](#)
- [PoissonGaussianNoise2DObjective.cpp](#)

## 9.45 mappel::PoissonNoise1DObjective Class Reference

A base class for 1D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/PoissonNoise1DObjective.h>
```

Inheritance diagram for mappel::PoissonNoise1DObjective:



### Public Types

- using `ModelDataT` = `ImageT`
- using `ModelDataStackT` = `ImageStackT`
- using `ImageCoordT` = `uint32_t`
- using `ImagePixelT` = `double`
- template<class CoordT >  
using `ImageSizeShapeT` = `CoordT`
- template<class CoordT >  
using `ImageSizeVecShapeT` = `arma::Col< CoordT >`
- using `ImageSizeT` = `ImageSizeShapeT< ImageCoordT >`
- using `ImageSizeVecT` = `ImageSizeVecShapeT< ImageCoordT >`
- template<class PixelT >  
using `ImageShapeT` = `arma::Col< PixelT >`
- template<class PixelT >  
using `ImageStackShapeT` = `arma::Mat< PixelT >`
- using `ImageT` = `ImageShapeT< ImagePixelT >`
- using `ImageStackT` = `ImageStackShapeT< ImagePixelT >`

### Public Member Functions

- `StatsT get_stats ()` const
- `ImageT make_image ()` const
- `ImageStackT make_image_stack (ImageCoordT n)` const
- `ImageCoordT get_size_image_stack (const ImageStackT &stack)` const
- `ImageT get_image_from_stack (const ImageStackT &stack, ImageCoordT n)` const
- template<class ImT >  
void `set_image_in_stack (ImageStackT &stack, ImageCoordT n, const ImT &im)` const
- `ImageSizeT get_size ()` const
- `ImageCoordT get_size (IdxT idx)` const
- `ImageCoordT get_num_pixels ()` const
- void `set_size (const ImageSizeT &size_)`
- void `set_size (const arma::Col< ImageCoordT > &sz)`
- void `check_image_shape (const ImageT &im)` const  
Check the shape of a single images is correct for model size.
- void `check_image_shape (const ImageStackT &ims)` const  
Check the shape of a stack of images is correct for model size.

### Static Public Member Functions

- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)  
*Check the size argument for the model.*

### Static Public Attributes

- static const std::vector< std::string > [estimator\\_names](#)
- static const [ImageCoordT](#) num\_dim = 1
- static const [ImageCoordT](#) global\_min\_size = 3
- static const [ImageCoordT](#) global\_max\_size = 512

### Protected Member Functions

- [PoissonNoise1DObjective](#) ()
- [PoissonNoise1DObjective](#) (const [PoissonNoise1DObjective](#) &o)
- [PoissonNoise1DObjective](#) ([PoissonNoise1DObjective](#) &&o)
- [PoissonNoise1DObjective](#) & operator= (const [PoissonNoise1DObjective](#) &o)
- [PoissonNoise1DObjective](#) & operator= ([PoissonNoise1DObjective](#) &&o)

### Protected Attributes

- [ImageSizeT](#) size

#### 9.45.1 Detailed Description

A base class for 1D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel.

Definition at line 22 of file [PoissonNoise1DObjective.h](#).

#### 9.45.2 Member Typedef Documentation

##### 9.45.2.1 using [mappel::ImageFormat1DBase::ImageCoordT](#) = [uint32\\_t](#) [inherited]

Image size coordinate storage type

Definition at line 25 of file [ImageFormat1DBase.h](#).

##### 9.45.2.2 using [mappel::ImageFormat1DBase::ImagePixelT](#) = [double](#) [inherited]

Image pixel storage type

Definition at line 26 of file [ImageFormat1DBase.h](#).

**9.45.2.3** `template<class PixelT > using mappel::ImageFormat1DBase::ImageShapeT = arma::Col<PixelT>`  
[inherited]

Shape of the data type for a single image

Definition at line 33 of file ImageFormat1DBase.h.

**9.45.2.4** `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeShapeT = CoordT` [inherited]

Shape of the data type to store 1-image's coordinates

Definition at line 28 of file ImageFormat1DBase.h.

**9.45.2.5** `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`  
[inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

**9.45.2.6** `template<class CoordT > using mappel::ImageFormat1DBase::ImageSizeVecShapeT = arma::Col<CoordT>`  
[inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 29 of file ImageFormat1DBase.h.

**9.45.2.7** `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
[inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

**9.45.2.8** `template<class PixelT > using mappel::ImageFormat1DBase::ImageStackShapeT = arma::Mat<PixelT>`  
[inherited]

Shape of the data type for a sequence of images

Definition at line 34 of file ImageFormat1DBase.h.

**9.45.2.9** `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
[inherited]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

9.45.2.10 **using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>** [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

9.45.2.11 **using mappel::PoissonNoise1DObjective::ModelDataStackT = ImageStackT**

Objective function data stack type: 1D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 26 of file PoissonNoise1DObjective.h.

9.45.2.12 **using mappel::PoissonNoise1DObjective::ModelDataT = ImageT**

Objective function data type: 1D double precision image, gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise1DObjective.h.

### 9.45.3 Constructor & Destructor Documentation

9.45.3.1 **mappel::PoissonNoise1DObjective::PoissonNoise1DObjective ( )** [protected]

Definition at line 14 of file PoissonNoise1DObjective.cpp.

9.45.3.2 **mappel::PoissonNoise1DObjective::PoissonNoise1DObjective ( const PoissonNoise1DObjective & o )**  
[protected]

Definition at line 18 of file PoissonNoise1DObjective.cpp.

9.45.3.3 **mappel::PoissonNoise1DObjective::PoissonNoise1DObjective ( PoissonNoise1DObjective && o )**  
[protected]

Definition at line 22 of file PoissonNoise1DObjective.cpp.

### 9.45.4 Member Function Documentation

9.45.4.1 **void ImageFormat1DBase::check\_image\_shape ( const ImageT & im ) const** [inherited]

Check the shape of a single images is correct for model size.

Definition at line 59 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.



9.45.4.2 `void ImageFormat1DBase::check_image_shape ( const ImageStackT & ims ) const` `[inherited]`

Check the shape of a stack of images is correct for model size.

Definition at line 71 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.45.4.3 `void ImageFormat1DBase::check_size ( const ImageSizeT & size_ )` `[static], [inherited]`

Check the size argument for the model.

Definition at line 39 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::global\_max\_size, and mappel::ImageFormat1DBase::global\_min\_size.

Referenced by mappel::ImageFormat1DBase::ImageFormat1DBase(), and mappel::ImageFormat1DBase::set\_size().

9.45.4.4 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` `[inline], [inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

9.45.4.5 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_num_pixels ( ) const` `[inline], [inherited]`

Definition at line 82 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

9.45.4.6 `ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get_size ( ) const` `[inline], [inherited]`

Definition at line 71 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::get\_stats().

9.45.4.7 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size ( IdxT idx ) const` `[inherited]`

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.45.4.8 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack ( const ImageStackT & stack ) const` `[inline], [inherited]`

Definition at line 101 of file ImageFormat1DBase.h.

#### 9.45.4.9 StatsT ImageFormat1DBase::get\_stats ( ) const [inherited]

Definition at line 81 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::get\_num\_pixels(), mappel::ImageFormat1DBase::get\_size(), and mappel::ImageFormat1DBase::num\_dim.

Referenced by mappel::Gauss1DModel::get\_stats(), and mappel::Gauss1DsModel::get\_stats().

#### 9.45.4.10 ImageFormat1DBase::ImageT ImageFormat1DBase::make\_image ( ) const [inline],[inherited]

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

#### 9.45.4.11 ImageFormat1DBase::ImageStackT ImageFormat1DBase::make\_image\_stack ( ImageCoordT n ) const [inline],[inherited]

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

#### 9.45.4.12 PoissonNoise1DObjective & mappel::PoissonNoise1DObjective::operator= ( const PoissonNoise1DObjective & o ) [protected]

Definition at line 26 of file PoissonNoise1DObjective.cpp.

Referenced by mappel::Gauss1DMap::operator=(), mappel::Gauss1DMLE::operator=(), mappel::Gauss1DsMap::operator=(), and mappel::Gauss1DsMLE::operator=().

#### 9.45.4.13 PoissonNoise1DObjective & mappel::PoissonNoise1DObjective::operator= ( PoissonNoise1DObjective && o ) [protected]

Definition at line 31 of file PoissonNoise1DObjective.cpp.

#### 9.45.4.14 template<class ImT > void ImageFormat1DBase::set\_image\_in\_stack ( ImageStackT & stack, ImageCoordT n, const ImT & im ) const [inherited]

Definition at line 115 of file ImageFormat1DBase.h.

#### 9.45.4.15 void ImageFormat1DBase::set\_size ( const ImageSizeT & size\_ ) [inherited]

Definition at line 30 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::check\_size(), and mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::set\_size(), mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DsModel::set\_size().

9.45.4.16 `void ImageFormat1DBase::set_size ( const arma::Col< ImageCoordT > & sz )` `[inline]`, `[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

## 9.45.5 Member Data Documentation

9.45.5.1 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names` `[static]`

Definition at line 24 of file PoissonNoise1DObjective.h.

9.45.5.2 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static]`, `[inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

9.45.5.3 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static]`, `[inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

9.45.5.4 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static]`, `[inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

9.45.5.5 `ImageSizeT mappel::ImageFormat1DBase::size` `[protected]`, `[inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

The documentation for this class was generated from the following files:

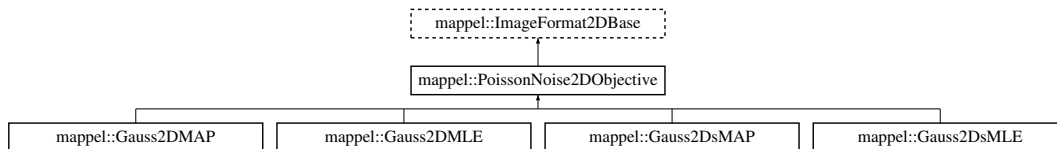
- [PoissonNoise1DObjective.h](#)
- [PoissonNoise1DObjective.cpp](#)

## 9.46 mappel::PoissonNoise2DObjective Class Reference

A base class for 2D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/PoissonNoise2DObjective.h>
```

Inheritance diagram for mappel::PoissonNoise2DObjective:



### Public Types

- using `ModelDataT` = `ImageT`
- using `ModelDataStackT` = `ImageStackT`
- using `ImageCoordT` = `uint32_t`
- using `ImagePixelT` = `double`
- template<class `CoordT` >  
using `ImageSizeShapeT` = `arma::Col< CoordT >`
- template<class `CoordT` >  
using `ImageSizeVecShapeT` = `arma::Mat< CoordT >`
- using `ImageSizeT` = `ImageSizeShapeT< ImageCoordT >`
- using `ImageSizeVecT` = `ImageSizeVecShapeT< ImageCoordT >`
- template<class `PixelT` >  
using `ImageShapeT` = `arma::Mat< PixelT >`
- template<class `PixelT` >  
using `ImageStackShapeT` = `arma::Cube< PixelT >`
- using `ImageT` = `ImageShapeT< ImagePixelT >`
- using `ImageStackT` = `ImageStackShapeT< ImagePixelT >`

### Public Member Functions

- `StatsT get_stats ()` const
- `ImageT make_image ()` const
- `ImageStackT make_image_stack (ImageCoordT n)` const
- `ImageCoordT get_size_image_stack (const ImageStackT &stack)` const
- `ImageT get_image_from_stack (const ImageStackT &stack, ImageCoordT n)` const
- template<class `ImT` >  
void `set_image_in_stack (ImageStackT &stack, ImageCoordT n, const ImT &im)` const
- const `ImageSizeT` & `get_size ()` const
- `ImageCoordT get_size (IdxT idx)` const
- `ImageCoordT get_num_pixels ()` const
- void `set_size (const ImageSizeT &size_)`
- void `check_image_shape (const ImageT &im)` const  
*Check the shape of a single images is correct for model size.*
- void `check_image_shape (const ImageStackT &ims)` const  
*Check the shape of a stack of images is correct for model size.*

### Static Public Member Functions

- static void [check\\_size](#) (const [ImageSizeT](#) &size\_)  
*Check the size argument for the model.*

### Static Public Attributes

- static const std::vector< std::string > [estimator\\_names](#)
- static const [ImageCoordT](#) [num\\_dim](#) =2
- static const [ImageCoordT](#) [global\\_min\\_size](#) =3
- static const [ImageCoordT](#) [global\\_max\\_size](#) =512

### Protected Member Functions

- [PoissonNoise2DObjective](#) ()
- [PoissonNoise2DObjective](#) (const [PoissonNoise2DObjective](#) &o)
- [PoissonNoise2DObjective](#) ([PoissonNoise2DObjective](#) &&o)
- [PoissonNoise2DObjective](#) & operator= (const [PoissonNoise2DObjective](#) &o)
- [PoissonNoise2DObjective](#) & operator= ([PoissonNoise2DObjective](#) &&o)

### Protected Attributes

- [ImageSizeT](#) [size](#)

#### 9.46.1 Detailed Description

A base class for 2D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel.

Definition at line 21 of file [PoissonNoise2DObjective.h](#).

#### 9.46.2 Member Typedef Documentation

##### 9.46.2.1 using mappel::ImageFormat2DBase::ImageCoordT = uint32\_t [inherited]

Image size coordinate storage type

Definition at line 24 of file [ImageFormat2DBase.h](#).

##### 9.46.2.2 using mappel::ImageFormat2DBase::ImagePixelT = double [inherited]

Image pixel storage type

Definition at line 25 of file [ImageFormat2DBase.h](#).

**9.46.2.3** `template<class PixelT > using mappel::ImageFormat2DBase::ImageShapeT = arma::Mat<PixelT>`  
[inherited]

Shape of the data type for a single image

Definition at line 32 of file ImageFormat2DBase.h.

**9.46.2.4** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeShapeT = arma::Col<CoordT>`  
[inherited]

Shape of the data type to store a single image's coordinates

Definition at line 27 of file ImageFormat2DBase.h.

**9.46.2.5** `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`  
[inherited]

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

**9.46.2.6** `template<class CoordT > using mappel::ImageFormat2DBase::ImageSizeVecShapeT = arma::Mat<CoordT>`  
[inherited]

Shape of the data type to store a vector of image's coordinates

Definition at line 28 of file ImageFormat2DBase.h.

**9.46.2.7** `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`  
[inherited]

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

**9.46.2.8** `template<class PixelT > using mappel::ImageFormat2DBase::ImageStackShapeT = arma::Cube<PixelT>`  
[inherited]

Shape of the data type for a sequence of images

Definition at line 33 of file ImageFormat2DBase.h.

**9.46.2.9** `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`  
[inherited]

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

9.46.2.10 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT> [inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

9.46.2.11 `using mappel::PoissonNoise2DObjective::ModelDataStackT = ImageStackT`

Objective function data stack type: 2D double precision image stack, of images gain-corrected to approximate photons counts

Definition at line 25 of file PoissonNoise2DObjective.h.

9.46.2.12 `using mappel::PoissonNoise2DObjective::ModelDataT = ImageT`

Objective function data type: 2D double precision image, gain-corrected to approximate photons counts

Definition at line 24 of file PoissonNoise2DObjective.h.

### 9.46.3 Constructor & Destructor Documentation

9.46.3.1 `mappel::PoissonNoise2DObjective::PoissonNoise2DObjective ( ) [protected]`

Definition at line 15 of file PoissonNoise2DObjective.cpp.

9.46.3.2 `mappel::PoissonNoise2DObjective::PoissonNoise2DObjective ( const PoissonNoise2DObjective & o ) [protected]`

Definition at line 19 of file PoissonNoise2DObjective.cpp.

9.46.3.3 `mappel::PoissonNoise2DObjective::PoissonNoise2DObjective ( PoissonNoise2DObjective && o ) [protected]`

Definition at line 23 of file PoissonNoise2DObjective.cpp.

### 9.46.4 Member Function Documentation

9.46.4.1 `void mappel::ImageFormat2DBase::check_image_shape ( const ImageT & im ) const [inherited]`

Check the shape of a single images is correct for model size.

Definition at line 80 of file ImageFormat2DBase.cpp.

References `mappel::ImageFormat2DBase::size`.

**9.46.4.2** `void mappel::ImageFormat2DBase::check_image_shape ( const ImageStackT & ims ) const` `[inherited]`

Check the shape of a stack of images is correct for model size.

Definition at line 93 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.46.4.3** `void mappel::ImageFormat2DBase::check_size ( const ImageSizeT & size_ )` `[static], [inherited]`

Check the size argument for the model.

Definition at line 60 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::global\_max\_size, and mappel::ImageFormat2DBase::global\_min\_size.

Referenced by mappel::ImageFormat2DBase::ImageFormat2DBase(), and mappel::ImageFormat2DBase::set\_size().

**9.46.4.4** `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get_image_from_stack ( const ImageStackT & stack, ImageCoordT n ) const` `[inline], [inherited]`

Definition at line 106 of file ImageFormat2DBase.h.

**9.46.4.5** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_num_pixels ( ) const` `[inline], [inherited]`

Definition at line 79 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

Referenced by mappel::ImageFormat2DBase::get\_stats().

**9.46.4.6** `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size ( ) const` `[inline], [inherited]`

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

**9.46.4.7** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size ( IdxT idx ) const` `[inherited]`

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

**9.46.4.8** `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack ( const ImageStackT & stack ) const` `[inline], [inherited]`

Definition at line 99 of file ImageFormat2DBase.h.



#### 9.46.4.9 StatsT mappel::ImageFormat2DBase::get\_stats ( ) const [inherited]

Definition at line 103 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::get\_num\_pixels(), mappel::ImageFormat2DBase::num\_dim, and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DModel::get\_stats(), and mappel::Gauss2DsModel::get\_stats().

#### 9.46.4.10 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make\_image ( ) const [inline], [inherited]

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

#### 9.46.4.11 ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make\_image\_stack ( ImageCoordT *n* ) const [inline], [inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

#### 9.46.4.12 PoissonNoise2DObjective & mappel::PoissonNoise2DObjective::operator= ( const PoissonNoise2DObjective & *o* ) [protected]

Definition at line 27 of file PoissonNoise2DObjective.cpp.

Referenced by mappel::Gauss2DsMAP::operator=(), mappel::Gauss2DsMLE::operator=(), mappel::Gauss2DMAP::operator=(), and mappel::Gauss2DMLE::operator=().

#### 9.46.4.13 PoissonNoise2DObjective & mappel::PoissonNoise2DObjective::operator= ( PoissonNoise2DObjective && *o* ) [protected]

Definition at line 32 of file PoissonNoise2DObjective.cpp.

#### 9.46.4.14 template<class ImT > void mappel::ImageFormat2DBase::set\_image\_in\_stack ( ImageStackT & *stack*, ImageCoordT *n*, const ImT & *im* ) const [inherited]

Definition at line 113 of file ImageFormat2DBase.h.

#### 9.46.4.15 void mappel::ImageFormat2DBase::set\_size ( const ImageSizeT & *size\_* ) [inherited]

Definition at line 51 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::check\_size(), and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DModel::set\_size(), and mappel::Gauss2DsModel::set\_size().

#### 9.46.5 Member Data Documentation

**9.46.5.1** `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` [static]

Definition at line 23 of file PoissonNoise2DObjective.h.

**9.46.5.2** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size =512` [static],  
[inherited]

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.46.5.3** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size =3` [static],  
[inherited]

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

**9.46.5.4** `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim =2` [static],  
[inherited]

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

**9.46.5.5** `ImageSizeT mappel::ImageFormat2DBase::size` [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by `mappel::cgauss_compute_estimate()`, `mappel::cgauss_compute_estimate_debug()`, `mappel::cgauss_←  
_heuristic_compute_estimate()`, `mappel::ImageFormat2DBase::check_image_shape()`, `mappel::Gauss2DModel::←  
Stencil::compute_derivatives()`, `mappel::Gauss2DsModel::Stencil::compute_derivatives()`, `mappel::Gauss2DModel::←  
::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::methods::←  
::expected_information()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsMAP::Gauss2DsMA←  
P()`, `mappel::Gauss2DsMLE::Gauss2DsMLE()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Image←  
Format2DBase::get_num_pixels()`, `mappel::ImageFormat2DBase::get_size()`, `mappel::ImageFormat2DBase::get←  
_stats()`, `mappel::methods::likelihood::grad()`, `mappel::methods::likelihood::grad2()`, `mappel::methods::likelihood←  
::debug::grad_components()`, `mappel::methods::likelihood::hessian()`, `mappel::methods::likelihood::debug::hessian←  
_components()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`,

mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh\_components(), mappel::Gauss2DModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_default\_prior\_beta\_position(), mappel::Gauss2DModel::make\_default\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_default\_prior\_normal\_position(), mappel::ImageFormat2DBase::make\_image(), mappel::ImageFormat2DBase::make\_image\_stack(), mappel::Gauss2DModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DsModel::make\_internal\_1Dsum\_estimator(), mappel::Gauss2DModel::make\_prior\_beta\_position(), mappel::Gauss2DsModel::make\_prior\_beta\_position(), mappel::Gauss2DModel::make\_prior\_normal\_position(), mappel::Gauss2DsModel::make\_prior\_normal\_position(), mappel::methods::model\_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods::likelihood::rllh(), mappel::methods::likelihood::debug::rllh\_components(), mappel::ImageFormat2DBase::set\_size(), mappel::Gauss2DModel::set\_size(), mappel::Gauss2DsModel::set\_size(), mappel::methods::simulate\_image(), mappel::methods::simulate\_image\_from\_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2DsModel::Stencil::Stencil(), mappel::Gauss2DModel::update\_internal\_1Dsum\_estimators(), and mappel::Gauss2DsModel::update\_internal\_1Dsum\_estimators().

The documentation for this class was generated from the following files:

- [PoissonNoise2DObjective.h](#)
- [PoissonNoise2DObjective.cpp](#)

## 9.47 mappel::estimator::ProfileBoundsData Struct Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

### Public Member Functions

- void [initialize\\_arrays](#) (IdxT Nparams)

### Public Attributes

- [IdxVecT estimated\\_idx](#)  
*List of indexes for computed parameters. Empty to compute all parameters.*
- double [confidence](#) = -1  
*Confidence level. If invalid, use default value.*
- [MLEData mle](#)  
*Theta maximum-likelihood estimate, rllh, and Obsl.*
- double [target\\_rllh\\_delta](#) = INFINITY  
*Targeted rllh change in value from MLE (  $-\chi^2_{inv}(\text{confidence}, 1)/2$  )*
- [IdxT Nparams\\_est](#) = 0  
*number of parameters estimated = estimated\_param\_idx.n\_elem.*
- [VecT profile\\_lb](#)  
*size:[Nparams\_est] Lower bound estimated at each estimated\_idx.*
- [VecT profile\\_ub](#)  
*size:[Nparams\_est] Upper bound estimated at each estimated\_idx.*
- [MatT profile\\_points\\_lb](#)  
*size:[NumParams, Nparams\_est] Optimal theta found at each lower bound estimate for each estimated\_idx.*
- [MatT profile\\_points\\_ub](#)  
*size:[NumParams, Nparams\_est] Optimal theta found at each upper bound estimate for each estimated\_idx.*
- [VecT profile\\_points\\_lb\\_rllh](#)  
*size:[Nparams\_est] RLLH at each of the profile\_points\_lb*
- [VecT profile\\_points\\_ub\\_rllh](#)  
*size:[Nparams\_est] RLLH at each of the profile\_points\_lb*

### 9.47.1 Detailed Description

Data related to a profile bounds estimation for a single image Includes both controlling (input) parameters as well as reporting (output) parameters to give output parameters context.

Definition at line 90 of file estimator.h.

### 9.47.2 Member Function Documentation

#### 9.47.2.1 void mappel::estimator::ProfileBoundsData::initialize\_arrays ( IdxT Nparams )

Definition at line 18 of file estimator.cpp.

References estimated\_idxxs, Nparams\_est, profile\_lb, profile\_points\_lb, profile\_points\_lb\_rllh, profile\_points\_ub, profile\_points\_ub\_rllh, and profile\_ub.

Referenced by mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_parallel(), and mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

### 9.47.3 Member Data Documentation

#### 9.47.3.1 double mappel::estimator::ProfileBoundsData::confidence =-1

Confidence level. If invalid, use default value.

Definition at line 94 of file estimator.h.

Referenced by mappel::methods::error\_bounds\_profile\_likelihood(), and mappel::methods::openmp::error\_bounds\_profile\_likelihood\_parallel().

#### 9.47.3.2 IdxVecT mappel::estimator::ProfileBoundsData::estimated\_idxxs

List of indexes for computed parameters. Empty to compute all parameters.

Definition at line 93 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::methods::error\_bounds\_profile\_likelihood(), mappel::methods::openmp::error\_bounds\_profile\_likelihood\_parallel(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_parallel(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack(), initialize\_arrays(), and mappel::estimator::ProfileBoundsDataStack::initialize\_arrays().

#### 9.47.3.3 MLEData mappel::estimator::ProfileBoundsData::mle

Theta maximum-likelihood estimate, rllh, and Obsl.

Definition at line 95 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_parallel(), and mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

**9.47.3.4 IdxT mappel::estimator::ProfileBoundsData::Nparams\_est =0**

number of parameters estimated =estimated\_param\_idx.n\_elem.

Definition at line 99 of file estimator.h.

Referenced by mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_parallel(), initialize\_arrays(), and mappel::estimator::ProfileBoundsDataStack::initialize\_arrays().

**9.47.3.5 VecT mappel::estimator::ProfileBoundsData::profile\_lb**

size:[Nparams\_est] Lower bound estimated at each estimated\_idx.

Definition at line 100 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack(), initialize\_arrays(), and mappel::estimator::ProfileBoundsDataStack::initialize\_arrays().

**9.47.3.6 MatT mappel::estimator::ProfileBoundsData::profile\_points\_lb**

size:[NumParams,Nparams\_est] Optimal theta found at each lower bound estimate for each estimated\_idx.

Definition at line 102 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack(), initialize\_arrays(), and mappel::estimator::ProfileBoundsDataStack::initialize\_arrays().

**9.47.3.7 VecT mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh**

size:[Nparams\_est] RLLH at each of the profile\_points\_lb

Definition at line 104 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack(), initialize\_arrays(), and mappel::estimator::ProfileBoundsDataStack::initialize\_arrays().

**9.47.3.8 MatT mappel::estimator::ProfileBoundsData::profile\_points\_ub**

size:[NumParams,Nparams\_est] Optimal theta found at each upper bound estimate for each estimated\_idx.

Definition at line 103 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack(), initialize\_arrays(), and mappel::estimator::ProfileBoundsDataStack::initialize\_arrays().

### 9.47.3.9 VecT mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh

size:[Nparams\_est] RLLH at each of the profile\_points\_lb

Definition at line 105 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack(), initialize\_arrays(), and mappel::estimator::ProfileBoundsDataStack::initialize\_arrays().

### 9.47.3.10 VecT mappel::estimator::ProfileBoundsData::profile\_ub

size:[Nparams\_est] Upper bound estimated at each estimated\_idx.

Definition at line 101 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack(), initialize\_arrays(), and mappel::estimator::ProfileBoundsDataStack::initialize\_arrays().

### 9.47.3.11 double mappel::estimator::ProfileBoundsData::target\_rllh\_delta =-INFINITY

Targeted rllh change in value from MLE (  $-\chi^2_{inv}(\text{confidence}, 1)/2$  )

Definition at line 98 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), mappel::methods::error\_bounds\_profile\_likelihood(), mappel::methods::openmp::error\_bounds\_profile\_likelihood\_parallel(), mappel::estimator::Estimator< Model >::estimate\_profile\_bounds(), mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_parallel(), and mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

The documentation for this struct was generated from the following files:

- [estimator.h](#)
- [estimator.cpp](#)

## 9.48 mappel::estimator::ProfileBoundsDataStack Struct Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

### Public Member Functions

- void [initialize\\_arrays](#) (IdxT Nparams)

## Public Attributes

- [IdxVecT estimated\\_idx](#)s  
*List of indexes for computed parameters. Empty to compute all parameters.*
- double [confidence](#) = -1  
*Confidence level. If invalid, use default value.*
- [MLEDataStack mle](#)  
*Theta maximum-likelihood estimate, rllh, and Obsl stack.*
- [IdxT Nparams\\_est](#) = 0  
*number of parameters estimated = estimated\_param\_idx.n\_elem.*
- [IdxT Ndata](#) = 0  
*size of the data stack estimated. (number of individual problem data estimates performed.)*
- double [target\\_rllh\\_delta](#) = -INFINITY  
*Targeted rllh change in value from MLE (  $-\chi^2_{inv}(\text{confidence}, 1)/2$  )*
- [MatT profile\\_lb](#)  
*size:[Nparams\_est, Ndata] Lower bound estimated at each estimated\_idx.*
- [MatT profile\\_ub](#)  
*size:[Nparams\_est, Ndata] Upper bound estimated at each estimated\_idx.*
- [CubeT profile\\_points\\_lb](#)  
*size:[Nparams, Nparams\_est, Ndata] Optimal theta found at each lower bound estimate for each estimated\_idx.*
- [CubeT profile\\_points\\_ub](#)  
*size:[Nparams, Nparams\_est, Ndata] Optimal theta found at each upper bound estimate for each estimated\_idx.*
- [MatT profile\\_points\\_lb\\_rllh](#)  
*size:[Nparams\_est, Ndata] RLLH at each of the profile\_points\_lb*
- [MatT profile\\_points\\_ub\\_rllh](#)  
*size:[Nparams\_est, Ndata] RLLH at each of the profile\_points\_ub*

## 9.48.1 Detailed Description

Data related to a profile bounds estimation for a stack of images Includes both controlling (input) parameters as well as reporting (ouput) parameters to give output parameters context.

Definition at line 134 of file estimator.h.

## 9.48.2 Member Function Documentation

9.48.2.1 void mappel::estimator::ProfileBoundsDataStack::initialize\_arrays ( [IdxT Nparams](#) )

Definition at line 29 of file estimator.cpp.

References [mappel::estimator::ProfileBoundsData::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsData::Nparams\\_est](#), [mappel::estimator::ProfileBoundsData::profile\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub\\_rllh](#), and [mappel::estimator::ProfileBoundsData::profile\\_ub](#).

Referenced by [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_stack\(\)](#).

### 9.48.3 Member Data Documentation

#### 9.48.3.1 `double mappel::estimator::ProfileBoundsDataStack::confidence = -1`

Confidence level. If invalid, use default value.

Definition at line 137 of file estimator.h.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_stack()`.

#### 9.48.3.2 `IdxVecT mappel::estimator::ProfileBoundsDataStack::estimated_idxxs`

List of indexs for computed parameters. Empty to compute all parameters.

Definition at line 136 of file estimator.h.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_stack()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

#### 9.48.3.3 `MLEDataStack mappel::estimator::ProfileBoundsDataStack::mle`

Theta maximum-likelihood estimate, rllh, and Obsl stack.

Definition at line 138 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

#### 9.48.3.4 `IdxT mappel::estimator::ProfileBoundsDataStack::Ndata = 0`

size of the data stack estimated. (number of individual problem data estimates performed.)

Definition at line 142 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

#### 9.48.3.5 `IdxT mappel::estimator::ProfileBoundsDataStack::Nparams_est = 0`

number of parameters estimated = `estimated_param_idxxs.n_elem`.

Definition at line 141 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

#### 9.48.3.6 `MatT mappel::estimator::ProfileBoundsDataStack::profile_lb`

size:[Nparams\_est,Ndata] Lower bound estimated at each `estimated_idx`.

Definition at line 144 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.



**9.48.3.7 CubeT mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb**

size:[Nparams,Nparams\_est,Ndata] Optimal theta found at each lower bound estimate for each estimated\_idx.

Definition at line 146 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

**9.48.3.8 MatT mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb\_rllh**

size:[Nparams\_est,Ndata] RLLH at each of the profile\_points\_lb

Definition at line 148 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

**9.48.3.9 CubeT mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub**

size:[Nparams,Nparams\_est,Ndata] Optimal theta found at each upper bound estimate for each estimated\_idx.

Definition at line 147 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

**9.48.3.10 MatT mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub\_rllh**

size:[Nparams\_est,Ndata] RLLH at each of the profile\_points\_ub

Definition at line 149 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

**9.48.3.11 MatT mappel::estimator::ProfileBoundsDataStack::profile\_ub**

size:[Nparams\_est,Ndata] Upper bound estimated at each estimated\_idx.

Definition at line 145 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

**9.48.3.12 double mappel::estimator::ProfileBoundsDataStack::target\_rllh\_delta =-INFINITY**

Targeted rllh change in value from MLE (  $-\chi^2_{\text{inv}}(\text{confidence},1)/2$  )

Definition at line 143 of file estimator.h.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_stack(), and mappel::estimator::ThreadedEstimator< Model >::estimate\_profile\_bounds\_stack().

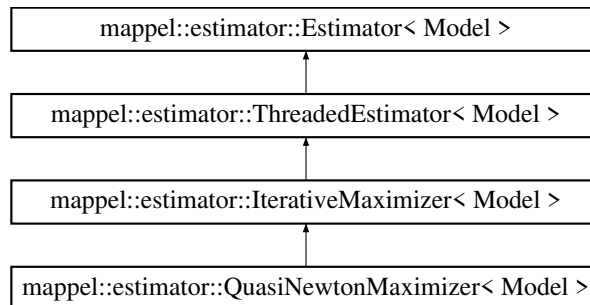
The documentation for this struct was generated from the following files:

- [estimator.h](#)
- [estimator.cpp](#)

### 9.49 mappel::estimator::QuasiNewtonMaximizer< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::estimator::QuasiNewtonMaximizer< Model >:



#### Public Types

- using [MaximizerData](#) = typename [IterativeMaximizer](#)< Model >::[MaximizerData](#)

#### Public Member Functions

- [QuasiNewtonMaximizer](#) (const Model &model, int max\_iterations=[IterativeMaximizer](#)< Model >::[DefaultIterations](#))
- std::string [name](#) () const
- double [mean\\_iterations](#) ()
- double [mean\\_backtracks](#) ()
- double [mean\\_fun\\_evals](#) ()
- double [mean\\_der\\_evals](#) ()
- [StatsT](#) [get\\_stats](#) ()
- [StatsT](#) [get\\_debug\\_stats](#) ()
- void [clear\\_stats](#) ()
- int [get\\_total\\_iterations](#) () const
- int [get\\_total\\_backtracks](#) () const
- int [get\\_total\\_fun\\_evals](#) () const
- int [get\\_total\\_der\\_evals](#) () const
- void [local\\_maximize](#) (const [ModelDataT](#)< Model > &im, [StencilT](#)< Model > &stencil, [MLEData](#) &data)  
*Perform a local maximization to finish off a simulated annealing run.*
- void [local\\_maximize](#) (const [ModelDataT](#)< Model > &im, [StencilT](#)< Model > &stencil, [MLEDebugData](#) &debug\_data)
- void [local\\_profile\\_maximize](#) (const [ModelDataT](#)< Model > &im, const [IdxVecT](#) &fixed\_param\_idx, [StencilT](#)< Model > &stencil, [MLEDebugData](#) &mle)
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [MLEDataStack](#) &mle\_data\_stack) override
- void [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [ProfileLikelihoodData](#) &profile) override
- void [estimate\\_profile\\_bounds\\_parallel](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est) override

- void `estimate_profile_bounds_stack` (const `ModelDataStackT`< Model > &data, `ProfileBoundsDataStack` &bounds\_est\_stack) override
- const Model & `get_model` ()
- void `estimate_max_stack` (const `ModelDataStackT`< Model > &data\_stack, `MLEDDataStack` &mle\_data\_stack)
- void `estimate_max` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil)
- void `estimate_max` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil)
- void `estimate_max` (const `ModelDataT`< Model > &data, `MLEDData` &mle\_data)
- void `estimate_max_debug` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil)
- void `estimate_max_debug` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data)
- double `estimate_profile_max` (const `ModelDataT`< Model > &data, const `IdxVecT` &fixed\_idxs, const `ParamT`< Model > &fixed\_theta\_init, `StencilT`< Model > &theta\_max)
- void `estimate_profile_bounds` (const `ModelDataT`< Model > &data, `ProfileBoundsData` &bounds\_est)
- void `estimate_profile_bounds_debug` (const `ModelDataT`< Model > &data, `ProfileBoundsDebugData` &bounds\_est)
- `IdxVecT` `get_exit_counts` () const

#### Static Public Attributes

- static const int `DefaultIterations` =100

#### Protected Member Functions

- void `record_run_statistics` (const `MaximizerData` &data)
- void `compute_estimate` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil) override
- void `compute_estimate_debug` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil) override
- double `compute_profile_estimate` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, const `IdxVecT` &fixed\_idxs, `StencilT`< Model > &theta\_max) override
- void `compute_profile_bound` (const `ModelDataT`< Model > &data, `ProfileBoundsData` &est, const `VecT` &init\_step, `IdxT` param\_idx, `IdxT` which\_bound) override
- void `compute_profile_bound_debug` (const `ModelDataT`< Model > &data, `ProfileBoundsDebugData` &bounds) override
- bool `backtrack` (`MaximizerData` &data)
- bool `profile_bound_backtrack` (`MaximizerData` &data, `IdxT` fixed\_idx, double target\_rllh, double old\_fval, const `VecT` &fggrad)
- virtual void `maximize` (`MaximizerData` &data)=0
- virtual void `solve_profile_bound` (`MaximizerData` &data, `MLEDData` &mle, double llh\_delta, `IdxT` fixed\_idx, `IdxT` which\_bound)
- bool `convergence_test_grad_ratio` (const `VecT` &grad, double fval)
- bool `convergence_test_step_size` (const `VecT` &new\_theta, const `VecT` &old\_theta)
- void `record_exit_code` (`ExitCode` code) override
- void `record_walltime` (`ClockT`::time\_point start\_walltime, int num\_estimations)

### Protected Attributes

- int `max_iterations`
- int `total_iterations` = 0
- int `total_backtracks` = 0
- int `total_fun_evals` = 0
- int `total_der_evals` = 0
- `IdxVecT` `last_backtrack_idxxs`

*Debugging: Stores last set of backtrack\_idxxs when data.save\_seq==true.*

- int `max_threads`
- int `num_threads`
- `std::mutex` `mtx`
- const Model & `model`
- int `num_estimations` = 0
- double `total_walltime` = 0.
- `IdxVecT` `exit_counts`

### Static Protected Attributes

- static const double `min_eigenvalue_correction_delta` = 1e-3

*Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.*

- static const double `convergence_min_function_change_ratio` = 1.0e-9

*Convergence criteria: tolerance for function-value change.*

- static const double `convergence_min_step_size_ratio` = 1.0e-9

*Convergence criteria: tolerance of relative step size.*

- static const double `backtrack_min_ratio` = 0.05
- static const double `backtrack_max_ratio` = 0.50
- static const double `backtrack_min_linear_step_ratio` = 1e-3
- static const int `max_backtracks` = 8
- static const double `min_profile_bound_residual` = 1e-4

*Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.*

#### 9.49.1 Detailed Description

```
template<class Model>
class mappel::estimator::QuasiNewtonMaximizer< Model >
```

Definition at line 599 of file estimator.h.

#### 9.49.2 Member Typedef Documentation

9.49.2.1 `template<class Model > using mappel::estimator::QuasiNewtonMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 602 of file estimator.h.

## 9.49.3 Constructor &amp; Destructor Documentation

9.49.3.1 `template<class Model> mappel::estimator::QuasiNewtonMaximizer< Model >::QuasiNewtonMaximizer ( const Model & model, int max_iterations = IterativeMaximizer<Model>::DefaultIterations ) [inline]`

Definition at line 604 of file estimator.h.

## 9.49.4 Member Function Documentation

9.49.4.1 `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::backtrack ( MaximizerData & data ) [protected],[inherited]`

Definition at line 870 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio`, `mappel::clamp()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::MaxBacktracks`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::StepSize`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.49.4.2 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::clear_stats ( ) [virtual],[inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 848 of file estimator\_impl.h.

References `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

9.49.4.3 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::compute_estimate ( const ModelDataT< Model> & data, const ParamT< Model> & theta_init, MLEData & mle_data, StencilT< Model> & mle_stencil ) [override], [protected], [virtual], [inherited]`

Implements [mappel::estimator::Estimator< Model>](#).

Definition at line 1043 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.49.4.4 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug ( const ModelDataT< Model> & im, const ParamT< Model> & theta_init, MLEDebugData & mle_debug, StencilT< Model> & mle_stencil ) [override], [protected], [virtual], [inherited]`

Virtual `estimate_debug` interface

Estimators that produce a sequence of results (e.g. `IterativeEstimators`) can override this dummy debug implementation.

Reimplemented from [mappel::estimator::Estimator< Model>](#).

Definition at line 1057 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEDebugData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.49.4.5 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound ( const ModelDataT< Model> & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound ) [override], [protected], [virtual], [inherited]`

Reimplemented from [mappel::estimator::Estimator< Model>](#).

Definition at line 1088 of file `estimator_impl.h`.

References `mappel::estimator::ProfileBoundsData::estimated_idx`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model>::model`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_points_ub_rllh`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::IterativeMaximizer< Model>::solve_profile_bound()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

9.49.4.6 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds )` [override], [protected], [virtual], [inherited]

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1114 of file estimator\_impl.h.

References [mappel::estimator::ProfileBoundsDebugData::estimated\\_idx](#), [mappel::estimator::ProfileBoundsDebugData::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsDebugData::Nseq\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::Nseq\\_ub](#), [mappel::estimator::MLEData::obsl](#), [mappel::estimator::ProfileBoundsDebugData::profile\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::profile\\_ub](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_ub](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_ub\\_rllh](#), [mappel::estimator::IterativeMaximizer< Model >::solve\\_profile\\_bound\(\)](#), [mappel::estimator::subroutine::solve\\_profile\\_initial\\_step\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step](#), [mappel::estimator::ProfileBoundsDebugData::target\\_rllh\\_delta](#), and [mappel::estimator::MLEData::theta](#).

9.49.4.7 `template<class Model> double mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxs, StencilT< Model > & theta_max )` [override], [protected], [virtual], [inherited]

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1074 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_fixed\\_parameters\(\)](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

9.49.4.8 `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio ( const VecT & grad, double fval )` [protected], [inherited]

Definition at line 1015 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_function\\_change\\_ratio](#), [mappel::estimator::GradRatio](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::square\(\)](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.49.4.9 `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size ( const VecT & new_theta, const VecT & old_theta )` [protected], [inherited]

Definition at line 1027 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_step\\_size\\_ratio](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::estimator::StepSize](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.49.4.10 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`.

9.49.4.11 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.



9.49.4.12 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::estimate\\_max\(\)](#), and [mappel::estimator::Estimator< Model >::model](#).

9.49.4.13 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at theta\_init, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The sequence and sequence\_rllh parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including theta\_init and theta\_mle, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.49.4.14 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data )`  
`[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

9.49.4.15 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack )` `[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max_stack()`, and `mappel::estimator::Estimator< Model >::model`.

9.49.4.16 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack )` [override],[virtual],[inherited]

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEDataStack::Ndata`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDataStack::obsI`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDataStack::rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDataStack::theta`.

9.49.4.17 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est )` [inherited]

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

9.49.4.18 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

9.49.4.19 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.

9.49.4.20 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, and `mappel::estimator::ProfileBoundsDataStack::initialize_arrays()`.

mappel::estimator::ProfileBoundsData::mle, mappel::estimator::ProfileBoundsDataStack::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsDataStack::Ndata, mappel::estimator::ProfileBoundsDataStack::Nparams\_est, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::MLEData::obsI, mappel::estimator::MLEDataStack::obsI, mappel::print\_text\_image(), mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_ub, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, mappel::estimator::MLEDataStack::rllh, mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, mappel::estimator::ProfileBoundsDataStack::target\_rllh\_delta, mappel::estimator::MLEData::theta, and mappel::estimator::MLEDataStack::theta.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_stack().

**9.49.4.21** `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), and mappel::methods::objective::rllh().

**9.49.4.22** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override],[virtual],[inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::estimator::ProfileLikelihoodData::fixed\_idx, mappel::estimator::ProfileLikelihoodData::fixed\_values, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileLikelihoodData::Nfixed, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::ProfileLikelihoodData::Nvalues, mappel::print\_text\_image(), mappel::estimator::ProfileLikelihoodData::profile\_likelihood, mappel::estimator::ProfileLikelihoodData::profile\_parameters, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), and mappel::estimator::Estimator< Model >::record\_walltime().

**9.49.4.23** `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_debug_stats ( )`  
`[virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 832 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx`s.

**9.49.4.24** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts ( ) const`  
`[inline], [inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.49.4.25** `template<class Model > const Model & mappel::estimator::Estimator< Model >::get_model ( )`  
`[inherited]`

Definition at line 108 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

**9.49.4.26** `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_stats ( )`  
`[virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 811 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::Estimator< Model >::num_estimations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`, `mappel::methods::debug::error_bounds_profile_likelihood_debug()`, `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`, `mappel::methods::openmp::error_bounds_profile_likelihood_stack()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`.

9.49.4.27 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_backtracks ( ) const`  
`[inline], [inherited]`

Definition at line 441 of file estimator.h.

9.49.4.28 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals ( ) const`  
`[inline], [inherited]`

Definition at line 443 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.49.4.29 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals ( ) const`  
`[inline], [inherited]`

Definition at line 442 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.49.4.30 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_iterations ( ) const`  
`[inline], [inherited]`

Definition at line 440 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.49.4.31 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::local_maximize ( const`  
`ModelDataT< Model> & im, StencilT< Model> & stencil, MLEData & data ) [inherited]`

Perform a local maximization to finish off a simulated annealing run.

Definition at line 1145 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.49.4.32 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::local_maximize ( const`  
`ModelDataT< Model> & im, StencilT< Model> & stencil, MLEDebugData & debug_data ) [inherited]`

Definition at line 1158 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEDebugData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.49.4.33 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize ( const ModelDataT< Model> & im, const IdxVecT & fixed_param_idx, StencilT< Model> & stencil, MLEDebugData & mle ) [inherited]`

Definition at line 1173 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::backtrack()`, `mappel::estimator::subroutine::bound_step()`, `mappel::clamp()`, `mappel::estimator::subroutine::compute_bound_scaling_vec()`, `mappel::estimator::subroutine::compute_cauchy_point()`, `mappel::estimator::subroutine::compute_initial_trust_radius()`, `mappel::estimator::subroutine::compute_quadratic_model_value()`, `mappel::estimator::subroutine::compute_scaled_problem()`, `mappel::estimator::IterativeMaximizer< Model>::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model>::convergence_test_step_size()`, `mappel::estimator::Estimator< Model>::exit_counts`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::fixed_idx`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::free_idx`, `mappel::estimator::Estimator< Model>::get_exit_counts()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model>::get_total_der_evals()`, `mappel::estimator::IterativeMaximizer< Model>::get_total_fun_evals()`, `mappel::estimator::IterativeMaximizer< Model>::get_total_iterations()`, `mappel::methods::objective::grad()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::grad`, `mappel::methods::objective::grad2()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::has_fixed_parameters()`, `mappel::methods::objective::hessian()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::im`, `mappel::is_positive_definite()`, `mappel::estimator::IterativeMaximizer< Model>::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model>::max_iterations`, `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::MaxIter`, `mappel::estimator::IterativeMaximizer< Model>::min_profile_bound_residual`, `mappel::estimator::Estimator< Model>::model`, `mappel::estimator::ThreadedEstimator< Model>::mtx`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::num_fixed_parameters()`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obs`, `mappel::estimator::MLEDebugData::obs`, `mappel::estimator::IterativeMaximizer< Model>::profile_bound_backtrack()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::restore_stencil()`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::s0`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::s1`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::saved_theta()`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::set_fixed_parameters()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::set_stencil()`, `mappel::estimator::subroutine::solve_TR_subproblem()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::step`, `mappel::estimator::MLEData::theta`, `mappel::estimator::MLEDebugData::theta`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`, `mappel::estimator::IterativeMaximizer< Model>::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model>::total_fun_evals`, `mappel::estimator::IterativeMaximizer< Model>::total_iterations`, and `mappel::estimator::TrustRegionRadius`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

9.49.4.34 `template<class Model> virtual void mappel::estimator::IterativeMaximizer< Model>::maximize ( MaximizerData & data ) [protected],[pure virtual],[inherited]`

Referenced by `mappel::estimator::IterativeMaximizer< Model>::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.



9.49.4.35 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_backtracks ( )`  
[inherited]

9.49.4.36 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_der_evals ( )`  
[inherited]

9.49.4.37 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_fun_evals ( )`  
[inherited]

9.49.4.38 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_iterations ( )`  
[inherited]

9.49.4.39 `template<class Model > std::string mappel::estimator::QuasiNewtonMaximizer< Model >::name ( ) const`  
[inline],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 607 of file estimator.h.

9.49.4.40 `template<class Model > bool mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack ( MaximizerData & data, IdxT fixed_idx, double target_rllh, double old_fval, const VecT & fgrad )` [protected],  
[inherited]

Definition at line 943 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_max\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_linear\\_step\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_ratio](#), [mappel::clamp\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_step\\_size\\_ratio](#), [mappel::methods::objective::grad\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im](#), [mappel::estimator::IterativeMaximizer< Model >::max\\_backtracks](#), [mappel::estimator::MaxBacktracks](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_backtrack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_iteration\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore\\_stencil\(\)](#), [mappel::methods::objective::rllh\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save\\_stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved\\_theta\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_stencil\(\)](#), [mappel::square\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step](#), [mappel::estimator::StepSize](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.49.4.41 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code )` [override], [protected], [virtual], [inherited]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::Success`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.49.4.42 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::record_run_statistics ( const MaximizerData & data )` [protected], [inherited]

Definition at line 859 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idxs()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idxs`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nBacktracks`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nIterations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.49.4.43 `template<class Model > void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` [protected], [inherited]

Definition at line 360 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

9.49.4.44 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::solve_profile_bound ( MaximizerData & data, MLEData & mle, double llh_delta, IdxT fixed_idx, IdxT which_bound )` [protected], [virtual], [inherited]

Definition at line 1137 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::name().

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), and mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound\_debug().

#### 9.49.5 Member Data Documentation

9.49.5.1 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_max_ratio = 0.50` [static], [protected], [inherited]

Definition at line 460 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.49.5.2 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio = 1e-3` [static], [protected], [inherited]

Definition at line 461 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::backtrack(), and mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.49.5.3 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_ratio = 0.05` [static], [protected], [inherited]

Definition at line 459 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.49.5.4 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio = 1.0e-9` [static], [protected], [inherited]

Convergence criteria: tolerance for function-value change.

Definition at line 456 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::convergence\_test\_grad\_ratio(), and mappel::estimator::IterativeMaximizer< Model >::get\_stats().

**9.49.5.5** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio = 1.0e-9` [static], [protected], [inherited]

Convergence criteria: tolerance of relative step size.

Definition at line 457 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.49.5.6** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::DefaultIterations = 100` [static], [inherited]

Definition at line 428 of file estimator.h.

**9.49.5.7** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 299 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.49.5.8** `template<class Model > IdxVecT mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx` [protected], [inherited]

Debugging: Stores last set of `backtrack_idx`s when `data.save_seq==true`.

Definition at line 475 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.49.5.9** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::max_backtracks = 8` [static], [protected], [inherited]

Definition at line 462 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.49.5.10** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::max_iterations` [protected], [inherited]

Definition at line 466 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.49.5.11 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::max_threads`  
`[protected], [inherited]`

Definition at line 324 of file estimator.h.

9.49.5.12 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model`  
`>::min_eigenvalue_correction_delta = 1e-3 [static], [protected], [inherited]`

Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.

Definition at line 454 of file estimator.h.

9.49.5.13 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model`  
`>::min_profile_bound_residual = 1e-4 [static], [protected], [inherited]`

Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.

Definition at line 464 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.49.5.14 `template<class Model > const Model& mappel::estimator::Estimator< Model >::model` `[protected],`  
`[inherited]`

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

9.49.5.15 `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` `[protected],`  
`[inherited]`

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.49.5.16** `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected], [inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

**9.49.5.17** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.

**9.49.5.18** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_backtracks = 0` [protected], [inherited]

Definition at line 470 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.49.5.19** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_der_evals = 0` [protected], [inherited]

Definition at line 472 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.49.5.20** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_fun_evals = 0` [protected], [inherited]

Definition at line 471 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.49.5.21 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::total_iterations = 0`  
`[protected], [inherited]`

Definition at line 469 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.49.5.22 `template<class Model> double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected], [inherited]`

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

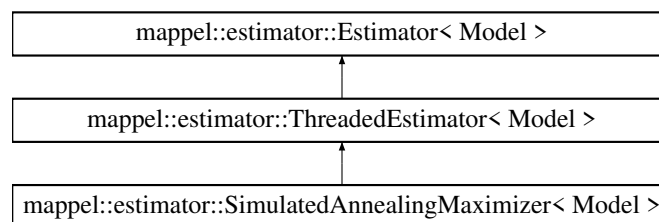
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.50 mappel::estimator::SimulatedAnnealingMaximizer< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::estimator::SimulatedAnnealingMaximizer< Model >`:



### Public Member Functions

- [SimulatedAnnealingMaximizer](#) (`const Model &model`, `int num_iterations_=DefaultNumIterations`, `double T_init_=Default\_T\_Init`, `double cooling_rate_=DefaultCoolingRate`)
- [StatsT get\\_stats](#) ()
- [StatsT get\\_debug\\_stats](#) ()
- `std::string name` () `const`
- `void estimate\_max\_stack (const ModelDataStackT< Model > &data, const ParamVecT< Model > &theta_init_←_stack, MLEDataStack &mle_data_stack) override`

- void `estimate_profile_max` (const `ModelDataT`< Model > &data, const `ParamVecT`< Model > &theta\_init, `ProfileLikelihoodData` &profile) override
  - void `estimate_profile_bounds_parallel` (const `ModelDataT`< Model > &data, `ProfileBoundsData` &bounds\_est) override
  - void `estimate_profile_bounds_stack` (const `ModelDataStackT`< Model > &data, `ProfileBoundsDataStack` &bounds\_est\_stack) override
  - void `clear_stats` ()
  - const Model & `get_model` ()
- 
- void `estimate_max_stack` (const `ModelDataStackT`< Model > &data\_stack, `MLEDDataStack` &mle\_data\_stack)
  - void `estimate_max` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil)
  - void `estimate_max` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data)
  - void `estimate_max` (const `ModelDataT`< Model > &data, `MLEDData` &mle\_data)
  - void `estimate_max_debug` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data, `StencilT`< Model > &mle\_stencil)
  - void `estimate_max_debug` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta\_init, `MLEDData` &mle\_data)
- 
- double `estimate_profile_max` (const `ModelDataT`< Model > &data, const `IdxVecT` &fixed\_idxes, const `ParamT`< Model > &fixed\_theta\_init, `StencilT`< Model > &theta\_max)
- 
- void `estimate_profile_bounds` (const `ModelDataT`< Model > &data, `ProfileBoundsData` &bounds\_est)
  - void `estimate_profile_bounds_debug` (const `ModelDataT`< Model > &data, `ProfileBoundsDebugData` &bounds\_est)
- 
- `IdxVecT` `get_exit_counts` () const

#### Static Public Attributes

- static const int `DefaultNumIterations` = 500  
*Default number of SA iterations.*
- static const double `Default_T_Init` = 100.  
*Default SA initial temperature.*
- static const double `DefaultCoolingRate` = 1.02  
*Default SA cooling rate.*

#### Protected Member Functions

- void `record_exit_code` (`ExitCode` code) override
- virtual void `compute_profile_bound` (const `ModelDataT`< Model > &data, `ProfileBoundsData` &est, const `VecT` &init\_step, `IdxT` param\_idx, `IdxT` which\_bound)
- virtual void `compute_profile_bound_debug` (const `ModelDataT`< Model > &data, `ProfileBoundsDebugData` &est)
- void `record_walltime` (`ClockT`::time\_point start\_walltime, int num\_estimations)



## Protected Attributes

- int [max\\_threads](#)
- int [num\\_threads](#)
- std::mutex [mtx](#)
- const Model & [model](#)
- int [num\\_estimations](#) = 0
- double [total\\_walltime](#) = 0.
- [IdxVecT](#) [exit\\_counts](#)

## 9.50.1 Detailed Description

```
template<class Model>
class mappel::estimator::SimulatedAnnealingMaximizer< Model >
```

Definition at line 384 of file estimator.h.

## 9.50.2 Constructor &amp; Destructor Documentation

9.50.2.1 `template<class Model > mappel::estimator::SimulatedAnnealingMaximizer< Model >::SimulatedAnnealingMaximizer ( const Model & model, int num_iterations = DefaultNumIterations, double T_init = Default_T_Init, double cooling_rate = DefaultCoolingRate ) [inline]`

Definition at line 392 of file estimator.h.

## 9.50.3 Member Function Documentation

9.50.3.1 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::clear_stats ( ) [virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 570 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::clear\\_stats\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::mtx](#), and [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::clear\\_stats\(\)](#).

9.50.3.2 `template<class Model > void mappel::estimator::Estimator< Model >::compute_profile_bound ( const ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 309 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

9.50.3.3 `template<class Model > void mappel::estimator::Estimator< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & est )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 318 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`.

9.50.3.4 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil )` [inherited]

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`,

mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, and mappel::estimator::MLEData::theta.

Referenced by mappel::estimator::Estimator< Model >::estimate\_max().

**9.50.3.5** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data )` [inherited]

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::estimate\_max().

**9.50.3.6** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data )` [inherited]

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::estimate\_max(), and mappel::estimator::Estimator< Model >::model.

**9.50.3.7** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at `theta_init`, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

**9.50.3.8** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

9.50.3.9 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::estimate\\_max\\_stack\(\)](#), and [mappel::estimator::Estimator< Model >::model](#).

9.50.3.10 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack ) [override],[virtual],[inherited]`

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::MLEDataStack::Ndata](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::MLEData::obsI](#), [mappel::estimator::MLEDataStack::obsI](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), [mappel::estimator::MLEData::rllh](#), [mappel::estimator::MLEDataStack::rllh](#), [mappel::estimator::MLEData::theta](#), and [mappel::estimator::MLEDataStack::theta](#).

9.50.3.11 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

9.50.3.12 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

9.50.3.13 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.

9.50.3.14 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_bound\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::ProfileBoundsData::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsDataStack::estimated\\_idx](#)s, [mappel::estimator::ProfileBoundsData::initialize\\_arrays\(\)](#), [mappel::estimator::ProfileBoundsDataStack::initialize\\_arrays\(\)](#), [mappel::estimator::ProfileBoundsData::mle](#), [mappel::estimator::ProfileBoundsDataStack::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsDataStack::Ndata](#), [mappel::estimator::ProfileBoundsDataStack::Nparams\\_est](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::MLEData::obs](#), [mappel::estimator::MLEDataStack::obs](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ProfileBoundsData::profile\\_lb](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_lb](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_ub](#), [mappel::estimator::ProfileBoundsData::profile\\_points\\_ub\\_rllh](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_points\\_ub\\_rllh](#), [mappel::estimator::ProfileBoundsData::profile\\_ub](#), [mappel::estimator::ProfileBoundsDataStack::profile\\_ub](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), [mappel::estimator::MLEData::rllh](#), [mappel::estimator::MLEDataStack::rllh](#), [mappel::estimator::subroutine::solve\\_profile\\_initial\\_step\(\)](#), [mappel::estimator::ProfileBoundsData::target\\_rllh\\_delta](#), [mappel::estimator::ProfileBoundsDataStack::target\\_rllh\\_delta](#), [mappel::estimator::MLEData::theta](#), and [mappel::estimator::MLEDataStack::theta](#).

Referenced by [mappel::methods::openmp::error\\_bounds\\_profile\\_likelihood\\_stack\(\)](#).

9.50.3.15 `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::Estimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

9.50.3.16 `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override], [virtual], [inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References [mappel::estimator::Estimator< Model >::compute\\_profile\\_estimate\(\)](#), [mappel::estimator::Error](#), [mappel::estimator::ProfileLikelihoodData::fixed\\_idx](#)s, [mappel::estimator::ProfileLikelihoodData::fixed\\_values](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileLikelihoodData::Nfixed](#), [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#), [mappel::estimator::ProfileLikelihoodData::Nvalues](#), [mappel::print\\_text\\_image\(\)](#), [mappel::estimator::ProfileLikelihoodData::profile\\_likelihood](#), [mappel::estimator::ProfileLikelihoodData::profile\\_parameters](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::estimator::Estimator< Model >::record\\_walltime\(\)](#).

9.50.3.17 `template<class Model> StatsT mappel::estimator::SimulatedAnnealingMaximizer< Model>::get_debug_stats ( ) [virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model>](#).

Definition at line 1764 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model>::get_stats()`.

9.50.3.18 `template<class Model> IdxVecT mappel::estimator::Estimator< Model>::get_exit_counts ( ) const [inline],[inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`.

9.50.3.19 `template<class Model> const Model & mappel::estimator::Estimator< Model>::get_model ( ) [inherited]`

Definition at line 108 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model>::model`.

9.50.3.20 `template<class Model> StatsT mappel::estimator::SimulatedAnnealingMaximizer< Model>::get_stats ( ) [virtual]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model>](#).

Definition at line 1744 of file `estimator_impl.h`.

References `mappel::estimator::ThreadedEstimator< Model>::get_stats()`, `mappel::estimator::ThreadedEstimator< Model>::mtx`, `mappel::estimator::Estimator< Model>::num_estimations`, `mappel::estimator::IterativeMaximizer< Model>::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model>::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model>::total_iterations`.

9.50.3.21 `template<class Model> std::string mappel::estimator::SimulatedAnnealingMaximizer< Model>::name ( ) const [inline],[virtual]`

Implements [mappel::estimator::Estimator< Model>](#).

Definition at line 401 of file `estimator.h`.

References `mappel::estimator::MLEData::rllh`.



9.50.3.22 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code )` [override], [protected], [virtual], [inherited]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::Success`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

9.50.3.23 `template<class Model > void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` [protected], [inherited]

Definition at line 360 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

#### 9.50.4 Member Data Documentation

9.50.4.1 `template<class Model > const double mappel::estimator::SimulatedAnnealingMaximizer< Model >::Default_T_Init = 100.` [static]

Default SA initial temperature.

Definition at line 389 of file `estimator.h`.

9.50.4.2 `template<class Model > const double mappel::estimator::SimulatedAnnealingMaximizer< Model >::DefaultCoolingRate = 1.02` [static]

Default SA cooling rate.

Definition at line 390 of file `estimator.h`.

**9.50.4.3** `template<class Model> const int mappel::estimator::SimulatedAnnealingMaximizer< Model >::DefaultNumIterations = 500 [static]`

Default number of SA iterations.

Definition at line 388 of file estimator.h.

**9.50.4.4** `template<class Model> IdxVecT mappel::estimator::Estimator< Model>::exit_counts [protected], [inherited]`

Definition at line 299 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model>::clear_stats()`, `mappel::estimator::Estimator< Model>::get_stats()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`.

**9.50.4.5** `template<class Model> int mappel::estimator::ThreadedEstimator< Model>::max_threads [protected], [inherited]`

Definition at line 324 of file estimator.h.

**9.50.4.6** `template<class Model> const Model& mappel::estimator::Estimator< Model>::model [protected], [inherited]`

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::backtrack()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate()`, `mappel::estimator::Estimator< Model>::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::estimator::Estimator< Model>::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound()`, `mappel::estimator::Estimator< Model>::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model>::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::estimator::Estimator< Model>::estimate_max()`, `mappel::estimator::Estimator< Model>::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::estimator::Estimator< Model>::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model>::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model>::get_debug_stats()`, `mappel::estimator::Estimator< Model>::get_model()`, `mappel::estimator::IterativeMaximizer< Model>::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model>::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`.

**9.50.4.7** `template<class Model> std::mutex mappel::estimator::ThreadedEstimator< Model>::mtx [protected], [inherited]`

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model>::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model>::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model>::get_stats()`, `mappel::estimator::HeuristicEstimator< Model>::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model>::get_stats()`, `mappel::estimator::CGaussMLE< Model>::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model>::get_stats()`, `mappel::estimator::IterativeMaximizer< Model>::get_stats()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`.

9.50.4.8 `template<class Model> int mappel::estimator::Estimator< Model>::num_estimations = 0` [protected],  
[inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model>::clear_stats()`, `mappel::estimator::Estimator< Model>::get_stats()`, `mappel::estimator::ThreadedEstimator< Model>::get_stats()`, `mappel::estimator::HeuristicEstimator< Model>::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model>::get_stats()`, `mappel::estimator::CGaussMLE< Model>::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model>::get_stats()`, `mappel::estimator::IterativeMaximizer< Model>::get_stats()`, and `mappel::estimator::Estimator< Model>::record_walltime()`.

9.50.4.9 `template<class Model> int mappel::estimator::ThreadedEstimator< Model>::num_threads` [protected],  
[inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model>::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model>::get_stats()`.

9.50.4.10 `template<class Model> double mappel::estimator::Estimator< Model>::total_walltime = 0.` [protected],  
[inherited]

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model>::clear_stats()`, `mappel::estimator::Estimator< Model>::get_stats()`, `mappel::estimator::ThreadedEstimator< Model>::get_stats()`, and `mappel::estimator::Estimator< Model>::record_walltime()`.

The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.51 mappel::Gauss2DsxyModel::Stencil Class Reference

[Stencil](#) for 2D free-sigma (astigmatic) models.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DsxyModel.h>
```

### Public Types

- `typedef Gauss2DsxyModel::ParamT` [ParamT](#)

**Public Member Functions**

- [Stencil](#) ()
- [Stencil](#) (const [Gauss2DsxxyModel](#) &[model](#), const [ParamT](#) &[theta](#), bool [\\_compute\\_derivatives](#)=true)
- void [compute\\_derivatives](#) ()
- double [x](#) () const
- double [y](#) () const
- double [l](#) () const
- double [bg](#) () const
- double [sigmaX](#) () const
- double [sigmaY](#) () const

**Public Attributes**

- bool [derivatives\\_computed](#) =false
- [Gauss2DsxxyModel](#) const \* [model](#)
- [ParamT](#) [theta](#)
- [VecT](#) [dx](#)
- [VecT](#) [dy](#)
- [VecT](#) [Gx](#)
- [VecT](#) [Gy](#)
- [VecT](#) [X](#)
- [VecT](#) [Y](#)
- [VecT](#) [DX](#)
- [VecT](#) [DY](#)
- [VecT](#) [DXSX](#)
- [VecT](#) [DYSX](#)
- [VecT](#) [DXS](#)
- [VecT](#) [DYS](#)
- [VecT](#) [DXS2](#)
- [VecT](#) [DYS2](#)
- [VecT](#) [DYSY](#)

**Friends**

- std::ostream & [operator<<](#) (std::ostream &out, const [Gauss2DsxxyModel::Stencil](#) &s)

**9.51.1 Detailed Description**

[Stencil](#) for 2D free-sigma (astigmatic) models.

Definition at line 32 of file [Gauss2DsxxyModel.h](#).

**9.51.2 Member Typedef Documentation****9.51.2.1 typedef [Gauss2DsxxyModel::ParamT](#) [mappel::Gauss2DsxxyModel::Stencil::ParamT](#)**

Definition at line 35 of file [Gauss2DsxxyModel.h](#).

### 9.51.3 Constructor & Destructor Documentation

#### 9.51.3.1 mappel::Gauss2DsxxyModel::Stencil ( ) [inline]

Definition at line 47 of file Gauss2DsxxyModel.h.

References `compute_derivatives()`.

Referenced by `mappel::Gauss2DsxxyModel::make_stencil()`.

#### 9.51.3.2 mappel::Gauss2DsxxyModel::Stencil ( const Gauss2DsxxyModel & model, const ParamT & theta, bool \_compute\_derivatives = true )

### 9.51.4 Member Function Documentation

#### 9.51.4.1 double mappel::Gauss2DsxxyModel::Stencil::bg ( ) const [inline]

Definition at line 53 of file Gauss2DsxxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxxyModel::pixel_model_value()`.

#### 9.51.4.2 void mappel::Gauss2DsxxyModel::Stencil::compute\_derivatives ( )

Referenced by `Stencil()`.

#### 9.51.4.3 double mappel::Gauss2DsxxyModel::Stencil::l ( ) const [inline]

Definition at line 52 of file Gauss2DsxxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad()`, `mappel::Gauss2DsxxyModel::pixel_grad2()`, `mappel::Gauss2DsxxyModel::pixel_hess()`, and `mappel::Gauss2DsxxyModel::pixel_model_value()`.

#### 9.51.4.4 double mappel::Gauss2DsxxyModel::Stencil::sigmaX ( ) const [inline]

Definition at line 54 of file Gauss2DsxxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad2()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

#### 9.51.4.5 double mappel::Gauss2DsxxyModel::Stencil::sigmaY ( ) const [inline]

Definition at line 55 of file Gauss2DsxxyModel.h.

References `operator<<`, and `theta`.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad2()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

9.51.4.6 `double mappel::Gauss2DsxyModel::Stencil::x ( ) const [inline]`

Definition at line 50 of file Gauss2DsxyModel.h.

References theta.

9.51.4.7 `double mappel::Gauss2DsxyModel::Stencil::y ( ) const [inline]`

Definition at line 51 of file Gauss2DsxyModel.h.

References theta.

### 9.51.5 Friends And Related Function Documentation

9.51.5.1 `std::ostream& operator<< ( std::ostream & out, const Gauss2DsxyModel::Stencil & s ) [friend]`

Referenced by sigmaY().

### 9.51.6 Member Data Documentation

9.51.6.1 `bool mappel::Gauss2DsxyModel::Stencil::derivatives_computed =false`

Definition at line 34 of file Gauss2DsxyModel.h.

9.51.6.2 `VecT mappel::Gauss2DsxyModel::Stencil::dx`

Definition at line 39 of file Gauss2DsxyModel.h.

9.51.6.3 `VecT mappel::Gauss2DsxyModel::Stencil::DX`

Definition at line 42 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel\_grad(), and mappel::Gauss2DsxyModel::pixel\_hess().

9.51.6.4 `VecT mappel::Gauss2DsxyModel::Stencil::DXS`

Definition at line 44 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel\_grad(), mappel::Gauss2DsxyModel::pixel\_grad2(), and mappel::Gauss2DsxyModel::pixel\_hess().

9.51.6.5 `VecT mappel::Gauss2DsxyModel::Stencil::DXS2`

Definition at line 45 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel\_grad2(), and mappel::Gauss2DsxyModel::pixel\_hess().

**9.51.6.6 VecT mappel::Gauss2DsxxyModel::Stencil::DXSX**

Definition at line 43 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel\_hess().

**9.51.6.7 VecT mappel::Gauss2DsxxyModel::Stencil::dy**

Definition at line 39 of file Gauss2DsxxyModel.h.

**9.51.6.8 VecT mappel::Gauss2DsxxyModel::Stencil::DY**

Definition at line 42 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel\_grad(), and mappel::Gauss2DsxxyModel::pixel\_hess().

**9.51.6.9 VecT mappel::Gauss2DsxxyModel::Stencil::DYS**

Definition at line 44 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel\_grad(), mappel::Gauss2DsxxyModel::pixel\_grad2(), and mappel::Gauss2DsxxyModel::pixel\_hess().

**9.51.6.10 VecT mappel::Gauss2DsxxyModel::Stencil::DYS2**

Definition at line 45 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel\_grad2(), and mappel::Gauss2DsxxyModel::pixel\_hess().

**9.51.6.11 VecT mappel::Gauss2DsxxyModel::Stencil::DYSX**

Definition at line 43 of file Gauss2DsxxyModel.h.

**9.51.6.12 VecT mappel::Gauss2DsxxyModel::Stencil::DYSY**

Definition at line 46 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel\_hess().

**9.51.6.13 VecT mappel::Gauss2DsxxyModel::Stencil::Gx**

Definition at line 40 of file Gauss2DsxxyModel.h.

**9.51.6.14 VecT mappel::Gauss2DsxxyModel::Stencil::Gy**

Definition at line 40 of file Gauss2DsxxyModel.h.

#### 9.51.6.15 **Gauss2DsxxyModel** const\* mappel::Gauss2DsxxyModel::Stencil::model

Definition at line 36 of file Gauss2DsxxyModel.h.

#### 9.51.6.16 **ParamT** mappel::Gauss2DsxxyModel::Stencil::theta

Definition at line 38 of file Gauss2DsxxyModel.h.

Referenced by `bg()`, `l()`, `sigmaX()`, `sigmaY()`, `x()`, and `y()`.

#### 9.51.6.17 **VecT** mappel::Gauss2DsxxyModel::Stencil::X

Definition at line 41 of file Gauss2DsxxyModel.h.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad()`, `mappel::Gauss2DsxxyModel::pixel_grad2()`, `mappel::Gauss2DsxxyModel::pixel_hess()`, and `mappel::Gauss2DsxxyModel::pixel_model_value()`.

#### 9.51.6.18 **VecT** mappel::Gauss2DsxxyModel::Stencil::Y

Definition at line 41 of file Gauss2DsxxyModel.h.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad()`, `mappel::Gauss2DsxxyModel::pixel_grad2()`, `mappel::Gauss2DsxxyModel::pixel_hess()`, and `mappel::Gauss2DsxxyModel::pixel_model_value()`.

The documentation for this class was generated from the following file:

- [Gauss2DsxxyModel.h](#)

## 9.52 mappel::Gauss1DsModel::Stencil Class Reference

[Stencil](#) for 1D variable-sigma models.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DsModel.h>
```

### Public Types

- using [ParamT](#) = Gauss1DsModel::ParamT

### Public Member Functions

- [Stencil](#) ()
- [Stencil](#) (const [Gauss1DsModel](#) &model, const [ParamT](#) &theta, bool \_compute\_derivatives=true)
- void [compute\\_derivatives](#) ()
- double [x](#) () const
- double [l](#) () const
- double [bg](#) () const
- double [sigma](#) () const



## Public Attributes

- bool [derivatives\\_computed](#) = false
- [Gauss1DsModel](#) const \* [model](#)
- [ParamT](#) [theta](#)
- [VecT](#) [dx](#)
- [VecT](#) [Gx](#)
- [VecT](#) [X](#)
- [VecT](#) [DX](#)
- [VecT](#) [DXS](#)
- [VecT](#) [DXS2](#)
- [VecT](#) [DXSX](#)

## Friends

- std::ostream & [operator<<](#) (std::ostream &out, const [Gauss1DsModel::Stencil](#) &s)

## 9.52.1 Detailed Description

[Stencil](#) for 1D variable-sigma models.

Definition at line 24 of file [Gauss1DsModel.h](#).

## 9.52.2 Member Typedef Documentation

## 9.52.2.1 using mappel::Gauss1DsModel::Stencil::ParamT = Gauss1DsModel::ParamT

Definition at line 27 of file [Gauss1DsModel.h](#).

## 9.52.3 Constructor &amp; Destructor Documentation

## 9.52.3.1 mappel::Gauss1DsModel::Stencil::Stencil ( ) [inline]

Definition at line 38 of file [Gauss1DsModel.h](#).

References [compute\\_derivatives\(\)](#).

Referenced by [mappel::Gauss1DsModel::make\\_stencil\(\)](#).

## 9.52.3.2 mappel::Gauss1DsModel::Stencil::Stencil ( const Gauss1DsModel &amp; model, const ParamT &amp; theta, bool \_compute\_derivatives = true )

Definition at line 159 of file [Gauss1DsModel.cpp](#).

References [compute\\_derivatives\(\)](#), [dx](#), [mappel::make\\_d\\_stencil\(\)](#), [mappel::make\\_X\\_stencil\(\)](#), [model](#), [sigma\(\)](#), [mappel::ImageFormat1DBase::size](#), [X](#), and [x\(\)](#).

#### 9.52.4 Member Function Documentation

##### 9.52.4.1 `double mappel::Gauss1DsModel::Stencil::bg ( ) const [inline]`

Definition at line 43 of file Gauss1DsModel.h.

References theta.

Referenced by `mappel::Gauss1DsModel::initial_theta_estimate()`, and `mappel::Gauss1DsModel::pixel_model_value()`.

##### 9.52.4.2 `void mappel::Gauss1DsModel::Stencil::compute_derivatives ( )`

Definition at line 170 of file Gauss1DsModel.cpp.

References `derivatives_computed`, `dx`, `DX`, `DXS`, `DXS2`, `DXSX`, `Gx`, `mappel::make_DX_stencil()`, `mappel::make_DX←S2_stencil()`, `mappel::make_DXS_stencil()`, `mappel::make_DXSX_stencil()`, `mappel::make_G_stencil()`, `model`, `sigma()`, and `mappel::ImageFormat1DBase::size`.

Referenced by `Stencil()`.

##### 9.52.4.3 `double mappel::Gauss1DsModel::Stencil::l ( ) const [inline]`

Definition at line 42 of file Gauss1DsModel.h.

References theta.

Referenced by `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::pixel_grad()`, `mappel←::Gauss1DsModel::pixel_grad2()`, `mappel::Gauss1DsModel::pixel_hess()`, `mappel::Gauss1DsModel::pixel_hess←update()`, and `mappel::Gauss1DsModel::pixel_model_value()`.

##### 9.52.4.4 `double mappel::Gauss1DsModel::Stencil::sigma ( ) const [inline]`

Definition at line 44 of file Gauss1DsModel.h.

References operator<<, and theta.

Referenced by `compute_derivatives()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss1DsModel←::pixel_grad2()`, `mappel::Gauss1DsModel::pixel_hess()`, `mappel::Gauss1DsModel::pixel_hess_update()`, and `Stencil()`.

##### 9.52.4.5 `double mappel::Gauss1DsModel::Stencil::x ( ) const [inline]`

Definition at line 41 of file Gauss1DsModel.h.

References theta.

Referenced by `Stencil()`.

### 9.52.5 Friends And Related Function Documentation

#### 9.52.5.1 `std::ostream& operator<< ( std::ostream & out, const Gauss1DsModel::Stencil & s )` [friend]

Definition at line 182 of file Gauss1DsModel.cpp.

Referenced by `sigma()`.

### 9.52.6 Member Data Documentation

#### 9.52.6.1 `bool mappel::Gauss1DsModel::Stencil::derivatives_computed = false`

Definition at line 26 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, and `mappel::operator<<()`.

#### 9.52.6.2 `VecT mappel::Gauss1DsModel::Stencil::dx`

Definition at line 31 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, and `Stencil()`.

#### 9.52.6.3 `VecT mappel::Gauss1DsModel::Stencil::DX`

Definition at line 34 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, `mappel::Gauss1DsModel::pixel_grad()`, `mappel::Gauss1DsModel::pixel_hess()`, and `mappel::Gauss1DsModel::pixel_hess_update()`.

#### 9.52.6.4 `VecT mappel::Gauss1DsModel::Stencil::DXS`

Definition at line 35 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, `mappel::Gauss1DsModel::pixel_grad()`, `mappel::Gauss1DsModel::pixel_grad2()`, `mappel::Gauss1DsModel::pixel_hess()`, and `mappel::Gauss1DsModel::pixel_hess_update()`.

#### 9.52.6.5 `VecT mappel::Gauss1DsModel::Stencil::DXS2`

Definition at line 36 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, `mappel::Gauss1DsModel::pixel_grad2()`, `mappel::Gauss1DsModel::pixel_hess()`, and `mappel::Gauss1DsModel::pixel_hess_update()`.

#### 9.52.6.6 `VecT mappel::Gauss1DsModel::Stencil::DXSX`

Definition at line 37 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, `mappel::Gauss1DsModel::pixel_hess()`, and `mappel::Gauss1DsModel::pixel_hess_update()`.

#### 9.52.6.7 **VecT** mappel::Gauss1DsModel::Stencil::Gx

Definition at line 32 of file Gauss1DsModel.h.

Referenced by compute\_derivatives(), and mappel::operator<<().

#### 9.52.6.8 **Gauss1DsModel** const\* mappel::Gauss1DsModel::Stencil::model

Definition at line 28 of file Gauss1DsModel.h.

Referenced by compute\_derivatives(), and Stencil().

#### 9.52.6.9 **ParamT** mappel::Gauss1DsModel::Stencil::theta

Definition at line 30 of file Gauss1DsModel.h.

Referenced by bg(), l(), mappel::operator<<(), sigma(), and x().

#### 9.52.6.10 **VecT** mappel::Gauss1DsModel::Stencil::X

Definition at line 33 of file Gauss1DsModel.h.

Referenced by mappel::operator<<(), mappel::Gauss1DsModel::pixel\_grad(), mappel::Gauss1DsModel::pixel\_↵ model\_value(), and Stencil().

The documentation for this class was generated from the following files:

- [Gauss1DsModel.h](#)
- [Gauss1DsModel.cpp](#)

### 9.53 mappel::Gauss2DModel::Stencil Class Reference

**Stencil** for 2D fixed-sigma models.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DModel.h>
```

#### Public Types

- using **ParamT** = [Gauss2DModel::ParamT](#)

#### Public Member Functions

- [Stencil](#) ()
- [Stencil](#) (const [Gauss2DModel](#) &model, const [ParamT](#) &theta, bool compute\_derivatives=true)
- void [compute\\_derivatives](#) ()
- double [x](#) () const
- double [y](#) () const
- double [l](#) () const
- double [bg](#) () const

### Public Attributes

- bool [derivatives\\_computed](#) =false
- [Gauss2DModel](#) const \* [model](#)
- [ParamT](#) [theta](#)
- [VecT](#) [dx](#)
- [VecT](#) [dy](#)
- [VecT](#) [Gx](#)
- [VecT](#) [Gy](#)
- [VecT](#) [X](#)
- [VecT](#) [Y](#)
- [VecT](#) [DX](#)
- [VecT](#) [DY](#)
- [VecT](#) [DXS](#)
- [VecT](#) [DYS](#)

### Friends

- std::ostream & [operator<<](#) (std::ostream &out, const [Gauss2DModel::Stencil](#) &s)

#### 9.53.1 Detailed Description

[Stencil](#) for 2D fixed-sigma models.

Definition at line 26 of file [Gauss2DModel.h](#).

#### 9.53.2 Member Typedef Documentation

##### 9.53.2.1 using mappel::Gauss2DModel::Stencil::ParamT = Gauss2DModel::ParamT

Definition at line 29 of file [Gauss2DModel.h](#).

#### 9.53.3 Constructor & Destructor Documentation

##### 9.53.3.1 mappel::Gauss2DModel::Stencil::Stencil ( ) [inline]

Definition at line 38 of file [Gauss2DModel.h](#).

References [compute\\_derivatives\(\)](#).

Referenced by [mappel::Gauss2DModel::make\\_stencil\(\)](#).

**9.53.3.2** `mappel::Gauss2DModel::Stencil::Stencil ( const Gauss2DModel & model, const ParamT & theta, bool compute_derivatives = true )`

Definition at line 218 of file Gauss2DModel.cpp.

References `compute_derivatives()`, `dx`, `dy`, `mappel::make_d_stencil()`, `mappel::make_X_stencil()`, `model`, `mappel::Gauss2DModel::psf_sigma`, `mappel::ImageFormat2DBase::size`, `X`, `x()`, `Y`, and `y()`.

#### 9.53.4 Member Function Documentation

**9.53.4.1** `double mappel::Gauss2DModel::Stencil::bg ( ) const [inline]`

Definition at line 44 of file Gauss2DModel.h.

References `operator<<`, and `theta`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, and `mappel::Gauss2DModel::pixel_model_value()`.

**9.53.4.2** `void mappel::Gauss2DModel::Stencil::compute_derivatives ( )`

Definition at line 232 of file Gauss2DModel.cpp.

References `derivatives_computed`, `dx`, `DX`, `DXS`, `dy`, `DY`, `DYS`, `Gx`, `Gy`, `mappel::make_DX_stencil()`, `mappel::make_DXS_stencil()`, `mappel::make_G_stencil()`, `model`, `mappel::Gauss2DModel::psf_sigma`, and `mappel::ImageFormat2DBase::size`.

Referenced by `Stencil()`.

**9.53.4.3** `double mappel::Gauss2DModel::Stencil::l ( ) const [inline]`

Definition at line 43 of file Gauss2DModel.h.

References `theta`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::pixel_grad()`, `mappel::Gauss2DModel::pixel_grad2()`, `mappel::Gauss2DModel::pixel_hess()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DModel::pixel_model_value()`.

**9.53.4.4** `double mappel::Gauss2DModel::Stencil::x ( ) const [inline]`

Definition at line 41 of file Gauss2DModel.h.

References `theta`.

Referenced by `Stencil()`.

9.53.4.5 `double mappel::Gauss2DModel::Stencil::y ( ) const [inline]`

Definition at line 42 of file Gauss2DModel.h.

References `theta`.

Referenced by `Stencil()`.

### 9.53.5 Friends And Related Function Documentation

9.53.5.1 `std::ostream& operator<< ( std::ostream & out, const Gauss2DModel::Stencil & s ) [friend]`

Definition at line 249 of file Gauss2DModel.cpp.

Referenced by `bg()`.

### 9.53.6 Member Data Documentation

9.53.6.1 `bool mappel::Gauss2DModel::Stencil::derivatives_computed =false`

Definition at line 28 of file Gauss2DModel.h.

Referenced by `compute_derivatives()`, and `mappel::operator<<()`.

9.53.6.2 `VecT mappel::Gauss2DModel::Stencil::dx`

Definition at line 33 of file Gauss2DModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, and `Stencil()`.

9.53.6.3 `VecT mappel::Gauss2DModel::Stencil::DX`

Definition at line 36 of file Gauss2DModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, `mappel::Gauss2DModel::pixel_grad()`, `mappel::Gauss2DModel::pixel_hess()`, and `mappel::Gauss2DModel::pixel_hess_update()`.

9.53.6.4 `VecT mappel::Gauss2DModel::Stencil::DXS`

Definition at line 37 of file Gauss2DModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, `mappel::Gauss2DModel::pixel_grad2()`, `mappel::Gauss2DModel::pixel_hess()`, and `mappel::Gauss2DModel::pixel_hess_update()`.

**9.53.6.5 VecT mappel::Gauss2DModel::Stencil::dy**

Definition at line 33 of file Gauss2DModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), and Stencil().

**9.53.6.6 VecT mappel::Gauss2DModel::Stencil::DY**

Definition at line 36 of file Gauss2DModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DModel::pixel\_grad(), mappel::Gauss2DModel::pixel\_hess(), and mappel::Gauss2DModel::pixel\_hess\_update().

**9.53.6.7 VecT mappel::Gauss2DModel::Stencil::DYS**

Definition at line 37 of file Gauss2DModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DModel::pixel\_grad2(), mappel::Gauss2DModel::pixel\_hess(), and mappel::Gauss2DModel::pixel\_hess\_update().

**9.53.6.8 VecT mappel::Gauss2DModel::Stencil::Gx**

Definition at line 34 of file Gauss2DModel.h.

Referenced by compute\_derivatives(), and mappel::operator<<().

**9.53.6.9 VecT mappel::Gauss2DModel::Stencil::Gy**

Definition at line 34 of file Gauss2DModel.h.

Referenced by compute\_derivatives(), and mappel::operator<<().

**9.53.6.10 Gauss2DModel const\* mappel::Gauss2DModel::Stencil::model**

Definition at line 30 of file Gauss2DModel.h.

Referenced by compute\_derivatives(), and Stencil().

**9.53.6.11 ParamT mappel::Gauss2DModel::Stencil::theta**

Definition at line 32 of file Gauss2DModel.h.

Referenced by bg(), l(), mappel::operator<<(), x(), and y().

**9.53.6.12 VecT mappel::Gauss2DModel::Stencil::X**

Definition at line 35 of file Gauss2DModel.h.

Referenced by mappel::operator<<(), mappel::Gauss2DModel::pixel\_grad(), mappel::Gauss2DModel::pixel\_grad2(), mappel::Gauss2DModel::pixel\_hess(), mappel::Gauss2DModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_model\_value(), and Stencil().



## 9.53.6.13 VecT mappel::Gauss2DModel::Stencil::Y

Definition at line 35 of file Gauss2DModel.h.

Referenced by mappel::operator<<(), mappel::Gauss2DModel::pixel\_grad(), mappel::Gauss2DModel::pixel\_grad2(), mappel::Gauss2DModel::pixel\_hess(), mappel::Gauss2DModel::pixel\_hess\_update(), mappel::Gauss2DModel::pixel\_model\_value(), and Stencil().

The documentation for this class was generated from the following files:

- [Gauss2DModel.h](#)
- [Gauss2DModel.cpp](#)

## 9.54 mappel::Gauss2DsModel::Stencil Class Reference

[Stencil](#) for 2D scalar-sigma models.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DsModel.h>
```

## Public Types

- typedef [Gauss2DsModel::ParamT](#) ParamT

## Public Member Functions

- [Stencil](#) ()
- [Stencil](#) (const [Gauss2DsModel](#) &model, const [ParamT](#) &theta, bool \_compute\_derivatives=true)
- void [compute\\_derivatives](#) ()
- double [x](#) () const
- double [y](#) () const
- double [l](#) () const
- double [bg](#) () const
- double [sigma\\_ratio](#) () const
- double [sigmaX](#) () const
- double [sigmaY](#) () const

## Public Attributes

- bool [derivatives\\_computed](#) =false
- [Gauss2DsModel](#) const \* [model](#)
- [ParamT](#) [theta](#)
- [VecT](#) [dx](#)
- [VecT](#) [dy](#)
- [VecT](#) [Gx](#)
- [VecT](#) [Gy](#)
- [VecT](#) [X](#)
- [VecT](#) [Y](#)
- [VecT](#) [DX](#)
- [VecT](#) [DY](#)
- [VecT](#) [DXS](#)
- [VecT](#) [DYS](#)
- [VecT](#) [DXS2](#)
- [VecT](#) [DYS2](#)
- [VecT](#) [DXSX](#)
- [VecT](#) [DYSY](#)

## Friends

- `std::ostream & operator<< (std::ostream &out, const Gauss2DsModel::Stencil &s)`

### 9.54.1 Detailed Description

[Stencil](#) for 2D scalar-sigma models.

Definition at line 29 of file Gauss2DsModel.h.

### 9.54.2 Member Typedef Documentation

#### 9.54.2.1 `typedef Gauss2DsModel::ParamT mappel::Gauss2DsModel::Stencil::ParamT`

Definition at line 32 of file Gauss2DsModel.h.

### 9.54.3 Constructor & Destructor Documentation

#### 9.54.3.1 `mappel::Gauss2DsModel::Stencil::Stencil ( ) [inline]`

Definition at line 43 of file Gauss2DsModel.h.

References `compute_derivatives()`.

Referenced by `mappel::Gauss2DsModel::make_stencil()`.

#### 9.54.3.2 `mappel::Gauss2DsModel::Stencil::Stencil ( const Gauss2DsModel & model, const ParamT & theta, bool _compute_derivatives = true )`

Definition at line 282 of file Gauss2DsModel.cpp.

References `compute_derivatives()`, `dx`, `dy`, `mappel::make_d_stencil()`, `mappel::make_X_stencil()`, `model`, `sigmaX()`, `sigmaY()`, `mappel::ImageFormat2DBase::size`, `X`, `x()`, `Y`, and `y()`.

### 9.54.4 Member Function Documentation

#### 9.54.4.1 `double mappel::Gauss2DsModel::Stencil::bg ( ) const [inline]`

Definition at line 49 of file Gauss2DsModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::pixel_model_value()`.

## 9.54.4.2 void mappel::Gauss2DsModel::Stencil::compute\_derivatives ( )

Definition at line 296 of file Gauss2DsModel.cpp.

References derivatives\_computed, dx, DX, DXS, DXS2, DXSX, dy, DY, DYS, DYS2, DYSY, Gx, Gy, mappel::make\_←\_DX\_stencil(), mappel::make\_DXS2\_stencil(), mappel::make\_DXS\_stencil(), mappel::make\_DXSX\_stencil(), mappel::←make\_G\_stencil(), model, sigmaX(), sigmaY(), and mappel::ImageFormat2DBase::size.

Referenced by Stencil().

## 9.54.4.3 double mappel::Gauss2DsModel::Stencil::l ( ) const [inline]

Definition at line 48 of file Gauss2DsModel.h.

References theta.

Referenced by mappel::Gauss2DsModel::initial\_theta\_estimate(), mappel::Gauss2DsModel::pixel\_grad(), mappel::←Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), mappel::Gauss2DsModel::pixel\_hess\_←update(), and mappel::Gauss2DsModel::pixel\_model\_value().

## 9.54.4.4 double mappel::Gauss2DsModel::Stencil::sigma\_ratio ( ) const [inline]

Definition at line 50 of file Gauss2DsModel.h.

References theta.

Referenced by mappel::Gauss2DsModel::initial\_theta\_estimate(), sigmaX(), and sigmaY().

## 9.54.4.5 double mappel::Gauss2DsModel::Stencil::sigmaX ( ) const [inline]

Definition at line 51 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::min\_sigma, and sigma\_ratio().

Referenced by compute\_derivatives(), mappel::Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), mappel::Gauss2DsModel::pixel\_hess\_update(), and Stencil().

## 9.54.4.6 double mappel::Gauss2DsModel::Stencil::sigmaY ( ) const [inline]

Definition at line 52 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::min\_sigma, operator<<, and sigma\_ratio().

Referenced by compute\_derivatives(), mappel::Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), mappel::Gauss2DsModel::pixel\_hess\_update(), and Stencil().

## 9.54.4.7 double mappel::Gauss2DsModel::Stencil::x ( ) const [inline]

Definition at line 46 of file Gauss2DsModel.h.

References theta.

Referenced by Stencil().

#### 9.54.4.8 `double mappel::Gauss2DsModel::Stencil::y ( ) const [inline]`

Definition at line 47 of file Gauss2DsModel.h.

References theta.

Referenced by Stencil().

### 9.54.5 Friends And Related Function Documentation

#### 9.54.5.1 `std::ostream& operator<< ( std::ostream & out, const Gauss2DsModel::Stencil & s ) [friend]`

Definition at line 314 of file Gauss2DsModel.cpp.

Referenced by sigmaY().

### 9.54.6 Member Data Documentation

#### 9.54.6.1 `bool mappel::Gauss2DsModel::Stencil::derivatives_computed =false`

Definition at line 31 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), and mappel::operator<<().

#### 9.54.6.2 `VecT mappel::Gauss2DsModel::Stencil::dx`

Definition at line 36 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), and Stencil().

#### 9.54.6.3 `VecT mappel::Gauss2DsModel::Stencil::DX`

Definition at line 39 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DsModel::pixel\_grad(), mappel::Gauss2DsModel::pixel\_hess(), and mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.54.6.4 `VecT mappel::Gauss2DsModel::Stencil::DXS`

Definition at line 40 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DsModel::pixel\_grad(), mappel::Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), and mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.54.6.5 VecT mappel::Gauss2DsModel::Stencil::DXS2

Definition at line 41 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), and mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.54.6.6 VecT mappel::Gauss2DsModel::Stencil::DXSX

Definition at line 42 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DsModel::pixel\_hess(), and mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.54.6.7 VecT mappel::Gauss2DsModel::Stencil::dy

Definition at line 36 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), and Stencil().

#### 9.54.6.8 VecT mappel::Gauss2DsModel::Stencil::DY

Definition at line 39 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DsModel::pixel\_grad(), mappel::Gauss2DsModel::pixel\_hess(), and mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.54.6.9 VecT mappel::Gauss2DsModel::Stencil::DYS

Definition at line 40 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DsModel::pixel\_grad(), mappel::Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), and mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.54.6.10 VecT mappel::Gauss2DsModel::Stencil::DYS2

Definition at line 41 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), and mappel::Gauss2DsModel::pixel\_hess\_update().

#### 9.54.6.11 VecT mappel::Gauss2DsModel::Stencil::DYSY

Definition at line 42 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss2DsModel::pixel\_hess(), and mappel::Gauss2DsModel::pixel\_hess\_update().

**9.54.6.12 VecT mappel::Gauss2DsModel::Stencil::Gx**

Definition at line 37 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), and mappel::operator<<().

**9.54.6.13 VecT mappel::Gauss2DsModel::Stencil::Gy**

Definition at line 37 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), and mappel::operator<<().

**9.54.6.14 Gauss2DsModel const\* mappel::Gauss2DsModel::Stencil::model**

Definition at line 33 of file Gauss2DsModel.h.

Referenced by compute\_derivatives(), and Stencil().

**9.54.6.15 ParamT mappel::Gauss2DsModel::Stencil::theta**

Definition at line 35 of file Gauss2DsModel.h.

Referenced by bg(), l(), mappel::operator<<(), sigma\_ratio(), x(), and y().

**9.54.6.16 VecT mappel::Gauss2DsModel::Stencil::X**

Definition at line 38 of file Gauss2DsModel.h.

Referenced by mappel::operator<<(), mappel::Gauss2DsModel::pixel\_grad(), mappel::Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), mappel::Gauss2DsModel::pixel\_hess\_update(), mappel::Gauss2DsModel::pixel\_model\_value(), and Stencil().

**9.54.6.17 VecT mappel::Gauss2DsModel::Stencil::Y**

Definition at line 38 of file Gauss2DsModel.h.

Referenced by mappel::operator<<(), mappel::Gauss2DsModel::pixel\_grad(), mappel::Gauss2DsModel::pixel\_grad2(), mappel::Gauss2DsModel::pixel\_hess(), mappel::Gauss2DsModel::pixel\_hess\_update(), mappel::Gauss2DsModel::pixel\_model\_value(), and Stencil().

The documentation for this class was generated from the following files:

- [Gauss2DsModel.h](#)
- [Gauss2DsModel.cpp](#)

**9.55 mappel::Gauss1DModel::Stencil Class Reference**

[Stencil](#) for 1D fixed-sigma models.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DModel.h>
```

## Public Types

- using [ParamT](#) = [Gauss1DModel::ParamT](#)

## Public Member Functions

- [Stencil](#) ()
- [Stencil](#) (const [Gauss1DModel](#) &[model](#), const [ParamT](#) &[theta](#), bool [compute\\_derivatives](#)=true)
- void [compute\\_derivatives](#) ()
- double [x](#) () const
- double [l](#) () const
- double [bg](#) () const

## Public Attributes

- bool [derivatives\\_computed](#) = false
- [Gauss1DModel](#) const \* [model](#)
- [ParamT](#) [theta](#)
- [VecT](#) [dx](#)
- [VecT](#) [Gx](#)
- [VecT](#) [X](#)
- [VecT](#) [DX](#)
- [VecT](#) [DXS](#)

## Friends

- std::ostream & [operator<<](#) (std::ostream &out, const [Gauss1DModel::Stencil](#) &s)

### 9.55.1 Detailed Description

[Stencil](#) for 1D fixed-sigma models.

Definition at line 29 of file [Gauss1DModel.h](#).

### 9.55.2 Member Typedef Documentation

#### 9.55.2.1 using mappel::Gauss1DModel::Stencil::ParamT = Gauss1DModel::ParamT

Definition at line 32 of file [Gauss1DModel.h](#).

### 9.55.3 Constructor & Destructor Documentation

#### 9.55.3.1 `mappel::Gauss1DModel::Stencil::Stencil ( ) [inline]`

Definition at line 41 of file Gauss1DModel.h.

References `compute_derivatives()`.

Referenced by `mappel::Gauss1DModel::make_stencil()`.

#### 9.55.3.2 `mappel::Gauss1DModel::Stencil::Stencil ( const Gauss1DModel & model, const ParamT & theta, bool compute_derivatives = true )`

Definition at line 142 of file Gauss1DModel.cpp.

References `compute_derivatives()`, `dx`, `mappel::make_d_stencil()`, `mappel::make_X_stencil()`, `model`, `mappel::Gauss1DModel::psf_sigma`, `mappel::ImageFormat1DBase::size`, `X`, and `x()`.

### 9.55.4 Member Function Documentation

#### 9.55.4.1 `double mappel::Gauss1DModel::Stencil::bg ( ) const [inline]`

Definition at line 46 of file Gauss1DModel.h.

References `operator<<`, and `theta`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, and `mappel::Gauss1DModel::pixel_model_value()`.

#### 9.55.4.2 `void mappel::Gauss1DModel::Stencil::compute_derivatives ( )`

Definition at line 153 of file Gauss1DModel.cpp.

References `derivatives_computed`, `dx`, `DX`, `DXS`, `Gx`, `mappel::make_DX_stencil()`, `mappel::make_DXS_stencil()`, `mappel::make_G_stencil()`, `model`, `mappel::Gauss1DModel::psf_sigma`, and `mappel::ImageFormat1DBase::size`.

Referenced by `Stencil()`.

#### 9.55.4.3 `double mappel::Gauss1DModel::Stencil::l ( ) const [inline]`

Definition at line 45 of file Gauss1DModel.h.

References `theta`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_grad()`, `mappel::Gauss1DModel::pixel_grad2()`, `mappel::Gauss1DModel::pixel_hess()`, `mappel::Gauss1DModel::pixel_hess_update()`, and `mappel::Gauss1DModel::pixel_model_value()`.



#### 9.55.4.4 double mappel::Gauss1DModel::Stencil::x ( ) const [inline]

Definition at line 44 of file Gauss1DModel.h.

References theta.

Referenced by Stencil().

### 9.55.5 Friends And Related Function Documentation

#### 9.55.5.1 std::ostream& operator<< ( std::ostream & out, const Gauss1DModel::Stencil & s ) [friend]

Definition at line 164 of file Gauss1DModel.cpp.

Referenced by bg().

### 9.55.6 Member Data Documentation

#### 9.55.6.1 bool mappel::Gauss1DModel::Stencil::derivatives\_computed = false

Definition at line 31 of file Gauss1DModel.h.

Referenced by compute\_derivatives(), and mappel::operator<<().

#### 9.55.6.2 VecT mappel::Gauss1DModel::Stencil::dx

Definition at line 36 of file Gauss1DModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), and Stencil().

#### 9.55.6.3 VecT mappel::Gauss1DModel::Stencil::DX

Definition at line 39 of file Gauss1DModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss1DModel::pixel\_grad(), and mappel::Gauss1DModel::pixel\_hess().

#### 9.55.6.4 VecT mappel::Gauss1DModel::Stencil::DXS

Definition at line 40 of file Gauss1DModel.h.

Referenced by compute\_derivatives(), mappel::operator<<(), mappel::Gauss1DModel::pixel\_grad2(), mappel::Gauss1DModel::pixel\_hess(), and mappel::Gauss1DModel::pixel\_hess\_update().



## Public Member Functions

- [ThreadedEstimator](#) (const Model &model)
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [MLEDataStack](#) &mle\_data\_stack) override
- void [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamVecT](#)< Model > &theta\_init, [ProfileLikelihoodData](#) &profile) override
- void [estimate\\_profile\\_bounds\\_parallel](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est) override
- void [estimate\\_profile\\_bounds\\_stack](#) (const [ModelDataStackT](#)< Model > &data, [ProfileBoundsDataStack](#) &bounds\_est\_stack) override
- [StatsT](#) [get\\_stats](#) ()
- [StatsT](#) [get\\_debug\\_stats](#) ()
- void [clear\\_stats](#) ()
- virtual std::string [name](#) () const =0
- const Model & [get\\_model](#) ()
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data)
- void [estimate\\_max](#) (const [ModelDataT](#)< Model > &data, [MLEData](#) &mle\_data)
- void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEDebugData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- void [estimate\\_max\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEDebugData](#) &mle\_data)
- void [estimate\\_max\\_stack](#) (const [ModelDataStackT](#)< Model > &data\_stack, [MLEDataStack](#) &mle\_data\_stack)
- double [estimate\\_profile\\_max](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed\_idx, const [ParamT](#)< Model > &fixed\_theta\_init, [StencilT](#)< Model > &theta\_max)
- void [estimate\\_profile\\_bounds](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &bounds\_est)
- void [estimate\\_profile\\_bounds\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &bounds\_est)
- [IdxVecT](#) [get\\_exit\\_counts](#) () const

## Protected Member Functions

- void [record\\_exit\\_code](#) (ExitCode code) override
- virtual void [compute\\_estimate](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)=0
- virtual void [compute\\_estimate\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEDebugData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil)
- virtual double [compute\\_profile\\_estimate](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, const [IdxVecT](#) &fixed\_idx, [StencilT](#)< Model > &max\_stencil)
- virtual void [compute\\_profile\\_bound](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &est, const [VecT](#) &init\_step, [IdxT](#) param\_idx, [IdxT](#) which\_bound)
- virtual void [compute\\_profile\\_bound\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &est)
- void [record\\_walltime](#) (ClockT::time\_point start\_walltime, int num\_estimations)

## Protected Attributes

- int [max\\_threads](#)
- int [num\\_threads](#)
- std::mutex [mtx](#)
- const Model & [model](#)
- int [num\\_estimations](#) = 0
- double [total\\_walltime](#) = 0.
- [IdxVecT](#) [exit\\_counts](#)

### 9.56.1 Detailed Description

```
template<class Model>
class mappel::estimator::ThreadedEstimator< Model >
```

We avoid combining [Estimator](#) and [ThreadedEstimator](#) classes so that a future GPU implementation can inherit directly from [Estimator](#) as it will present a different method for `estimate_stack` pure virtual member function. For now all other (CPU) estimators inherit from [ThreadedEstimator](#).

Definition at line 309 of file `estimator.h`.

### 9.56.2 Constructor & Destructor Documentation

9.56.2.1 `template<class Model > mappel::estimator::ThreadedEstimator< Model >::ThreadedEstimator ( const Model & model )`

Definition at line 370 of file `estimator_impl.h`.

### 9.56.3 Member Function Documentation

9.56.3.1 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::clear_stats ( ) [virtual]`

Run statistics.

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 570 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::clear\\_stats\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::mtx](#), and [mappel::estimator::ThreadedEstimator< Model >::num\\_threads](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::clear\\_stats\(\)](#).

9.56.3.2 `template<class Model > virtual void mappel::estimator::Estimator< Model >::compute_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil )` [protected], [pure virtual], [inherited]

Implemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Referenced by `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`.

9.56.3.3 `template<class Model > void mappel::estimator::Estimator< Model >::compute_estimate_debug ( const ModelDataT< Model > & im, const ParamT< Model > & theta_init, MLEDebugData & mle_debug, StencilT< Model > & mle_stencil )` [protected], [virtual], [inherited]

Virtual `estimate_debug` interface

Estimators that produce a sequence of results (e.g. `IterativeEstimators`) can override this dummy debug implementation.

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 285 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDebugData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max_debug()`.

9.56.3.4 `template<class Model > void mappel::estimator::Estimator< Model >::compute_profile_bound ( const ModelDataT< Model > & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 309 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

9.56.3.5 `template<class Model > void mappel::estimator::Estimator< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & est )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 318 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`.

9.56.3.6 `template<class Model > double mappel::estimator::Estimator< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idx, StencilT< Model > & max_stencil )` [protected], [virtual], [inherited]

Reimplemented in [mappel::estimator::IterativeMaximizer< Model >](#).

Definition at line 300 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.

9.56.3.7 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil )` [inherited]

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`.

9.56.3.8 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data )` [inherited]

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.

**9.56.3.9** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::Estimator< Model >::model`.

**9.56.3.10** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at `theta_init`, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.56.3.11 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at *theta\_init*, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with *theta\_init*, *theta\_init* is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of *theta\_init* will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

9.56.3.12 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at *theta\_init*, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with *theta\_init*, *theta\_init* is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of *theta\_init* will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

## Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.



Definition at line 183 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max_stack()`, and `mappel::estimator::Estimator< Model >::model`.

**9.56.3.13** `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack ) [override], [virtual]`

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEDataStack::Ndata`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDataStack::obsI`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDataStack::rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDataStack::theta`.

**9.56.3.14** `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idxxs`, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

9.56.3.15 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est )` [inherited]

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

9.56.3.16 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est )` [override], [virtual]

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.

9.56.3.17 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est )` [override], [virtual]

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, and `mappel::estimator::ProfileBoundsDataStack::initialize_arrays()`.

mappel::estimator::ProfileBoundsData::mle, mappel::estimator::ProfileBoundsDataStack::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsDataStack::Ndata, mappel::estimator::ProfileBoundsDataStack::Nparams\_est, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::MLEData::obsI, mappel::estimator::MLEDataStack::obsI, mappel::print\_text\_image(), mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_ub, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, mappel::estimator::MLEDataStack::rllh, mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, mappel::estimator::ProfileBoundsDataStack::target\_rllh\_delta, mappel::estimator::MLEData::theta, and mappel::estimator::MLEDataStack::theta.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_stack().

**9.56.3.18** `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), and mappel::methods::objective::rllh().

**9.56.3.19** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override],[virtual]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::estimator::ProfileLikelihoodData::fixed\_idx, mappel::estimator::ProfileLikelihoodData::fixed\_values, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileLikelihoodData::Nfixed, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::ProfileLikelihoodData::Nvalues, mappel::print\_text\_image(), mappel::estimator::ProfileLikelihoodData::profile\_likelihood, mappel::estimator::ProfileLikelihoodData::profile\_parameters, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), and mappel::estimator::Estimator< Model >::record\_walltime().

**9.56.3.20** `template<class Model> StatsT mappel::estimator::ThreadedEstimator< Model>::get_debug_stats ( )`  
`[virtual]`

Run statistics.

Implements [mappel::estimator::Estimator< Model>](#).

Reimplemented in [mappel::estimator::IterativeMaximizer< Model>](#), [mappel::estimator::SimulatedAnnealingMaximizer< Model>](#), [mappel::estimator::CGaussMLE< Model>](#), [mappel::estimator::CGaussHeuristicEstimator< Model>](#), and [mappel::estimator::HeuristicEstimator< Model>](#).

Definition at line 564 of file `estimator_impl.h`.

References [mappel::estimator::ThreadedEstimator< Model>::get\\_stats\(\)](#).

**9.56.3.21** `template<class Model> IdxVecT mappel::estimator::Estimator< Model>::get_exit_counts ( ) const`  
`[inline],[inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by [mappel::estimator::IterativeMaximizer< Model>::local\\_profile\\_maximize\(\)](#).

**9.56.3.22** `template<class Model> const Model & mappel::estimator::Estimator< Model>::get_model ( )`  
`[inherited]`

Definition at line 108 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model>::model](#).

**9.56.3.23** `template<class Model> StatsT mappel::estimator::ThreadedEstimator< Model>::get_stats ( )`  
`[virtual]`

Run statistics.

Reimplemented from [mappel::estimator::Estimator< Model>](#).

Reimplemented in [mappel::estimator::IterativeMaximizer< Model>](#), [mappel::estimator::SimulatedAnnealingMaximizer< Model>](#), [mappel::estimator::CGaussMLE< Model>](#), [mappel::estimator::CGaussHeuristicEstimator< Model>](#), and [mappel::estimator::HeuristicEstimator< Model>](#).

Definition at line 553 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model>::get\\_stats\(\)](#), [mappel::estimator::ThreadedEstimator< Model>::mtx](#), [mappel::estimator::Estimator< Model>::num\\_estimations](#), [mappel::estimator::ThreadedEstimator< Model>::num\\_threads](#), and [mappel::estimator::Estimator< Model>::total\\_walltime](#).

Referenced by [mappel::estimator::ThreadedEstimator< Model>::get\\_debug\\_stats\(\)](#), [mappel::estimator::HeuristicEstimator< Model>::get\\_debug\\_stats\(\)](#), [mappel::estimator::CGaussHeuristicEstimator< Model>::get\\_debug\\_stats\(\)](#), [mappel::estimator::CGaussMLE< Model>::get\\_debug\\_stats\(\)](#), [mappel::estimator::HeuristicEstimator< Model>::get\\_stats\(\)](#), [mappel::estimator::CGaussHeuristicEstimator< Model>::get\\_stats\(\)](#), [mappel::estimator::CGaussMLE< Model>::get\\_stats\(\)](#), [mappel::estimator::SimulatedAnnealingMaximizer< Model>::get\\_stats\(\)](#), and [mappel::estimator::IterativeMaximizer< Model>::get\\_stats\(\)](#).

9.56.3.24 `template<class Model > virtual std::string mappel::estimator::Estimator< Model >::name ( ) const` [pure virtual],[inherited]

Implemented in [mappel::estimator::TrustRegionMaximizer< Model >](#), [mappel::estimator::QuasiNewtonMaximizer< Model >](#), [mappel::estimator::NewtonMaximizer< Model >](#), [mappel::estimator::NewtonDiagonalMaximizer< Model >](#), [mappel::estimator::SimulatedAnnealingMaximizer< Model >](#), [mappel::estimator::CGaussMLE< Model >](#), [mappel::estimator::CGaussHeuristicEstimator< Model >](#), and [mappel::estimator::HeuristicEstimator< Model >](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::solve\\_profile\\_bound\(\)](#).

9.56.3.25 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code )` [override],[protected],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::exit\\_counts](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ThreadedEstimator< Model >::mtx](#), [mappel::methods::observed\\_information\(\)](#), [mappel::estimator::MLEData::obsI](#), [mappel::estimator::MLEData::rllh](#), [mappel::methods::objective::rllh\(\)](#), [mappel::estimator::Success](#), and [mappel::estimator::MLEData::theta](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_test\\_grad\\_ratio\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_test\\_step\\_size\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_max\\_stack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_parallel\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_stack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_max\(\)](#), [mappel::estimator::HeuristicEstimator< Model >::get\\_debug\\_stats\(\)](#), [mappel::estimator::CGaussMLE< Model >::get\\_debug\\_stats\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#), and [mappel::estimator::IterativeMaximizer< Model >::profile\\_bound\\_backtrack\(\)](#).

9.56.3.26 `template<class Model > void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` [protected],[inherited]

Definition at line 360 of file `estimator_impl.h`.

References [mappel::estimator::Estimator< Model >::num\\_estimations](#), and [mappel::estimator::Estimator< Model >::total\\_walltime](#).

Referenced by [mappel::estimator::Estimator< Model >::estimate\\_max\(\)](#), [mappel::estimator::Estimator< Model >::estimate\\_max\\_debug\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_max\\_stack\(\)](#), [mappel::estimator::Estimator< Model >::estimate\\_profile\\_bounds\(\)](#), [mappel::estimator::Estimator< Model >::estimate\\_profile\\_bounds\\_debug\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_parallel\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_bounds\\_stack\(\)](#), [mappel::estimator::Estimator< Model >::estimate\\_profile\\_max\(\)](#), and [mappel::estimator::ThreadedEstimator< Model >::estimate\\_profile\\_max\(\)](#).

## 9.56.4 Member Data Documentation

9.56.4.1 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected],[inherited]

Definition at line 299 of file `estimator.h`.

Referenced by [mappel::estimator::Estimator< Model >::clear\\_stats\(\)](#), [mappel::estimator::Estimator< Model >::get\\_stats\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#), and [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#).

**9.56.4.2** `template<class Model> int mappel::estimator::ThreadedEstimator< Model>::max_threads` [protected]

Definition at line 324 of file estimator.h.

**9.56.4.3** `template<class Model> const Model& mappel::estimator::Estimator< Model>::model` [protected],  
[inherited]

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model>::backtrack()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate()`, `mappel::estimator::Estimator< Model>::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::estimator::Estimator< Model>::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound()`, `mappel::estimator::Estimator< Model>::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model>::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::estimator::Estimator< Model>::estimate_max()`, `mappel::estimator::Estimator< Model>::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::estimator::Estimator< Model>::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model>::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model>::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model>::get_debug_stats()`, `mappel::estimator::Estimator< Model>::get_model()`, `mappel::estimator::IterativeMaximizer< Model>::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model>::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`.

**9.56.4.4** `template<class Model> std::mutex mappel::estimator::ThreadedEstimator< Model>::mtx` [protected]

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model>::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model>::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model>::get_stats()`, `mappel::estimator::HeuristicEstimator< Model>::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model>::get_stats()`, `mappel::estimator::CGaussMLE< Model>::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model>::get_stats()`, `mappel::estimator::IterativeMaximizer< Model>::get_stats()`, `mappel::estimator::IterativeMaximizer< Model>::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`.

**9.56.4.5** `template<class Model> int mappel::estimator::Estimator< Model>::num_estimations = 0` [protected],  
[inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model>::clear_stats()`, `mappel::estimator::Estimator< Model>::get_stats()`, `mappel::estimator::ThreadedEstimator< Model>::get_stats()`, `mappel::estimator::HeuristicEstimator< Model>::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model>::get_stats()`, `mappel::estimator::CGaussMLE< Model>::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model>::get_stats()`, `mappel::estimator::IterativeMaximizer< Model>::get_stats()`, and `mappel::estimator::Estimator< Model>::record_walltime()`.

9.56.4.6 `template<class Model> int mappel::estimator::ThreadedEstimator< Model >::num_threads` `[protected]`

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.

9.56.4.7 `template<class Model> double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected]`, `[inherited]`

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

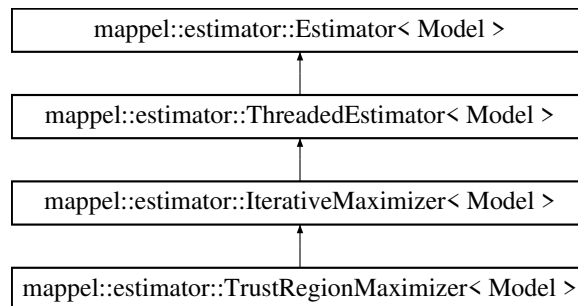
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator\\_impl.h](#)

## 9.57 mappel::estimator::TrustRegionMaximizer< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::estimator::TrustRegionMaximizer< Model >`:



### Public Types

- using `MaximizerData` = typename `IterativeMaximizer< Model >::MaximizerData`

## Public Member Functions

- [TrustRegionMaximizer](#) (const Model &model, int max\_iterations=IterativeMaximizer< Model >::DefaultIterations)
  - std::string name () const
  - double mean\_iterations ()
  - double mean\_backtracks ()
  - double mean\_fun\_evals ()
  - double mean\_der\_evals ()
  - StatsT get\_stats ()
  - StatsT get\_debug\_stats ()
  - void clear\_stats ()
  - int get\_total\_iterations () const
  - int get\_total\_backtracks () const
  - int get\_total\_fun\_evals () const
  - int get\_total\_der\_evals () const
  - void local\_maximize (const ModelDataT< Model > &im, StencilT< Model > &stencil, MLEData &data)
    - Perform a local maximization to finish off a simulated annealing run.*
  - void local\_maximize (const ModelDataT< Model > &im, StencilT< Model > &stencil, MLEDebugData &debug←\_data)
  - void local\_profile\_maximize (const ModelDataT< Model > &im, const IdxVecT &fixed\_param\_idx, StencilT< Model > &stencil, MLEDebugData &mle)
  - void estimate\_max\_stack (const ModelDataStackT< Model > &data, const ParamVecT< Model > &theta\_init←\_stack, MLEDataStack &mle\_data\_stack) override
  - void estimate\_profile\_max (const ModelDataT< Model > &data, const ParamVecT< Model > &theta\_init, ProfileLikelihoodData &profile) override
  - void estimate\_profile\_bounds\_parallel (const ModelDataT< Model > &data, ProfileBoundsData &bounds\_est) override
  - void estimate\_profile\_bounds\_stack (const ModelDataStackT< Model > &data, ProfileBoundsDataStack &bounds\_est\_stack) override
  - const Model & get\_model ()
- 
- void estimate\_max\_stack (const ModelDataStackT< Model > &data\_stack, MLEDataStack &mle\_data\_stack)
  - void estimate\_max (const ModelDataT< Model > &data, const ParamT< Model > &theta\_init, MLEData &mle←\_data, StencilT< Model > &mle\_stencil)
  - void estimate\_max (const ModelDataT< Model > &data, const ParamT< Model > &theta\_init, MLEData &mle←\_data)
  - void estimate\_max (const ModelDataT< Model > &data, MLEData &mle\_data)
  - void estimate\_max\_debug (const ModelDataT< Model > &data, const ParamT< Model > &theta\_init, MLE←\_DebugData &mle\_data, StencilT< Model > &mle\_stencil)
  - void estimate\_max\_debug (const ModelDataT< Model > &data, const ParamT< Model > &theta\_init, MLE←\_DebugData &mle\_data)
- 
- double estimate\_profile\_max (const ModelDataT< Model > &data, const IdxVecT &fixed\_idx, const ParamT< Model > &fixed\_theta\_init, StencilT< Model > &theta\_max)
- 
- void estimate\_profile\_bounds (const ModelDataT< Model > &data, ProfileBoundsData &bounds\_est)
  - void estimate\_profile\_bounds\_debug (const ModelDataT< Model > &data, ProfileBoundsDebugData &bounds←\_est)
- 
- IdxVecT get\_exit\_counts () const



## Static Public Attributes

- static const double [rho\\_cauchy\\_min](#) = 0.1  
*Minimum acceptable rho for cauchy point: Coleman beta / Bellavia beta\_1.*
- static const double [rho\\_obj\\_min](#) = 0.25  
*Minimum acceptable rho: Coleman mu / Bellavia beta\_2.*
- static const double [rho\\_obj\\_opt](#) = 0.75  
*Optimal step rho: Coleman eta / Bellavia beta\_2.*
- static const double [trust\\_radius\\_decrease\\_min](#) = 0.125  
*Smallest allowable trust radius decrease ratio: Coleman gamma\_0 / Bellavia alpha\_1.*
- static const double [trust\\_radius\\_decrease](#) = 0.25  
*Trust radius decrease ratio to step size: Coleman gamma\_1 / Bellavia alpha\_2.*
- static const double [trust\\_radius\\_increase](#) = 2  
*Trust radius increase ratio: Coleman gamma\_2 / Bellavia alpha\_3.*
- static const double [convergence\\_min\\_trust\\_radius](#) = 1.0e-8  
*Convergence criteria: Minimum trust region radius.*
- static const int [DefaultIterations](#) = 100

## Protected Member Functions

- void [record\\_run\\_statistics](#) (const [MaximizerData](#) &data)
- void [compute\\_estimate](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil) override
- void [compute\\_estimate\\_debug](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, [MLEDebugData](#) &mle\_data, [StencilT](#)< Model > &mle\_stencil) override
- double [compute\\_profile\\_estimate](#) (const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta\_init, const [IdxVecT](#) &fixed\_idx, [StencilT](#)< Model > &theta\_max) override
- void [compute\\_profile\\_bound](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsData](#) &est, const [VecT](#) &init\_← step, [IdxT](#) param\_idx, [IdxT](#) which\_bound) override
- void [compute\\_profile\\_bound\\_debug](#) (const [ModelDataT](#)< Model > &data, [ProfileBoundsDebugData](#) &bounds) override
- bool [backtrack](#) ([MaximizerData](#) &data)
- bool [profile\\_bound\\_backtrack](#) ([MaximizerData](#) &data, [IdxT](#) fixed\_idx, double target\_rllh, double old\_fval, const [VecT](#) &fgrad)
- virtual void [maximize](#) ([MaximizerData](#) &data)=0
- virtual void [solve\\_profile\\_bound](#) ([MaximizerData](#) &data, [MLEData](#) &mle, double llh\_delta, [IdxT](#) fixed\_idx, [IdxT](#) which\_bound)
- bool [convergence\\_test\\_grad\\_ratio](#) (const [VecT](#) &grad, double fval)
- bool [convergence\\_test\\_step\\_size](#) (const [VecT](#) &new\_theta, const [VecT](#) &old\_theta)
- void [record\\_exit\\_code](#) ([ExitCode](#) code) override
- void [record\\_walltime](#) ([ClockT](#)::time\_point start\_walltime, int [num\\_estimations](#))

### Protected Attributes

- int `max_iterations`
- int `total_iterations` = 0
- int `total_backtracks` = 0
- int `total_fun_evals` = 0
- int `total_der_evals` = 0
- `IdxVecT` `last_backtrack_idxxs`

*Debugging: Stores last set of backtrack\_idxxs when data.save\_seq==true.*

- int `max_threads`
- int `num_threads`
- `std::mutex` `mtx`
- const Model & `model`
- int `num_estimations` = 0
- double `total_walltime` = 0.
- `IdxVecT` `exit_counts`

### Static Protected Attributes

- static const double `min_eigenvalue_correction_delta` = 1e-3

*Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.*

- static const double `convergence_min_function_change_ratio` = 1.0e-9

*Convergence criteria: tolerance for function-value change.*

- static const double `convergence_min_step_size_ratio` = 1.0e-9

*Convergence criteria: tolerance of relative step size.*

- static const double `backtrack_min_ratio` = 0.05
- static const double `backtrack_max_ratio` = 0.50
- static const double `backtrack_min_linear_step_ratio` = 1e-3
- static const int `max_backtracks` = 8
- static const double `min_profile_bound_residual` = 1e-4

*Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.*

#### 9.57.1 Detailed Description

```
template<class Model>
class mappel::estimator::TrustRegionMaximizer< Model >
```

Definition at line 614 of file estimator.h.

#### 9.57.2 Member Typedef Documentation

9.57.2.1 `template<class Model> using mappel::estimator::TrustRegionMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 617 of file estimator.h.

## 9.57.3 Constructor &amp; Destructor Documentation

9.57.3.1 `template<class Model> mappel::estimator::TrustRegionMaximizer< Model >::TrustRegionMaximizer ( const Model & model, int max_iterations = IterativeMaximizer<Model>::DefaultIterations ) [inline]`

Definition at line 627 of file estimator.h.

## 9.57.4 Member Function Documentation

9.57.4.1 `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::backtrack ( MaximizerData & data ) [protected],[inherited]`

Definition at line 870 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio`, `mappel::clamp()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::MaxBacktracks`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::StepSize`, and `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.57.4.2 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::clear_stats ( ) [virtual],[inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 848 of file estimator\_impl.h.

References `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

9.57.4.3 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::compute_estimate ( const ModelDataT< Model> & data, const ParamT< Model> & theta_init, MLEData & mle_data, StencilT< Model> & mle_stencil ) [override], [protected], [virtual], [inherited]`

Implements [mappel::estimator::Estimator< Model>](#).

Definition at line 1043 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.57.4.4 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::compute_estimate_debug ( const ModelDataT< Model> & im, const ParamT< Model> & theta_init, MLEDebugData & mle_debug, StencilT< Model> & mle_stencil ) [override], [protected], [virtual], [inherited]`

Virtual `estimate_debug` interface

Estimators that produce a sequence of results (e.g. `IterativeEstimators`) can override this dummy debug implementation.

Reimplemented from [mappel::estimator::Estimator< Model>](#).

Definition at line 1057 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEDebugData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.57.4.5 `template<class Model> void mappel::estimator::IterativeMaximizer< Model>::compute_profile_bound ( const ModelDataT< Model> & data, ProfileBoundsData & est, const VecT & init_step, IdxT param_idx, IdxT which_bound ) [override], [protected], [virtual], [inherited]`

Reimplemented from [mappel::estimator::Estimator< Model>](#).

Definition at line 1088 of file estimator\_impl.h.

References `mappel::estimator::ProfileBoundsData::estimated_idx`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model>::model`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_points_ub_rllh`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::IterativeMaximizer< Model>::solve_profile_bound()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

9.57.4.6 `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds )` [override], [protected], [virtual], [inherited]

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1114 of file estimator\_impl.h.

References [mappel::estimator::ProfileBoundsDebugData::estimated\\_idx](#), [mappel::estimator::ProfileBoundsDebugData::mle](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::ProfileBoundsDebugData::Nseq\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::Nseq\\_ub](#), [mappel::estimator::MLEData::obsl](#), [mappel::estimator::ProfileBoundsDebugData::profile\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::profile\\_ub](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_lb](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_lb\\_rllh](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_ub](#), [mappel::estimator::ProfileBoundsDebugData::sequence\\_ub\\_rllh](#), [mappel::estimator::IterativeMaximizer< Model >::solve\\_profile\\_bound\(\)](#), [mappel::estimator::subroutine::solve\\_profile\\_initial\\_step\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step](#), [mappel::estimator::ProfileBoundsDebugData::target\\_rllh\\_delta](#), and [mappel::estimator::MLEData::theta](#).

9.57.4.7 `template<class Model> double mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxs, StencilT< Model > & theta_max )` [override], [protected], [virtual], [inherited]

Reimplemented from [mappel::estimator::Estimator< Model >](#).

Definition at line 1074 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::estimator::IterativeMaximizer< Model >::record\\_run\\_statistics\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_fixed\\_parameters\(\)](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

9.57.4.8 `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio ( const VecT & grad, double fval )` [protected], [inherited]

Definition at line 1015 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_function\\_change\\_ratio](#), [mappel::estimator::GradRatio](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::square\(\)](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.57.4.9 `template<class Model> bool mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size ( const VecT & new_theta, const VecT & old_theta )` [protected], [inherited]

Definition at line 1027 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_step\\_size\\_ratio](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), and [mappel::estimator::StepSize](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

9.57.4.10 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilT< Model > & mle_stencil )`  
`[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 128 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`.

9.57.4.11 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data )` `[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at theta\_init, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with theta\_init, theta\_init is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of theta\_init will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 121 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::estimate_max()`.

9.57.4.12 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max ( const ModelDataT< Model > & data, MLEData & mle_data ) [inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 112 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max()`, and `mappel::estimator::Estimator< Model >::model`.

9.57.4.13 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data, StencilT< Model > & mle_stencil ) [inherited]`

Debug estimation for a single data starting at `theta_init`, fill in the [MLEDebugData](#) struct with data including the sequence of evaluated points. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The sequence and `sequence_rllh` parameters of the [MLEDebugData](#) struct record the entire sequence of evaluated points including `theta_init` and `theta_mle`, which should be first and last respectively.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEDebugData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.57.4.14 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_debug ( const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEDebugData & mle_data )`  
`[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 157 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::Error`, `mappel::estimator::MLEDebugData::obsI`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEDebugData::rllh`, and `mappel::estimator::MLEDebugData::theta`.

9.57.4.15 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, MLEDataStack & mle_data_stack )` `[inherited]`

Maximum likelihood point estimators Estimate for a single data starting at `theta_init`, fill in the [MLEData](#) struct with the estimated parameter, RLLH, and observed information. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

The stencil at the MLE is also returned but can be ignored if not needed as it is available at no extra cost.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[Optional] Initial theta value.
out	<i>mle_data</i>	<a href="#">MLEData</a> recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

Definition at line 183 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::estimate_max_stack()`, and `mappel::estimator::Estimator< Model >::model`.



9.57.4.16 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack ( const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, MLEDataStack & mle_data_stack )` [override],[virtual],[inherited]

Estimate for a stack of data and fill in the [MLEDataStack](#) struct with the estimated parameter, RLLH, and observed information for each data in parallel. Estimation is initialized with `theta_init`, `theta_init` is empty, it is estimated with the Heuristic estimator. If any individual parameters are infinite or are not in the interior of the feasible region, they will be estimated with the Heuristic method. Valid parameters of `theta_init` will not be modified in the initialization process.

#### Parameters

in	<i>data</i>	Model data to estimate for
in	<i>theta_init</i>	[optional] Initial theta value for each image.
out	<i>mle</i>	MLEStackData records the maximum likelihood estimate, RLLH, and Observed information for each data

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 377 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_estimate()`, `mappel::estimator::Error`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::MLEDataStack::Ndata`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEDataStack::obsI`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::MLEDataStack::rllh`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::MLEDataStack::theta`.

9.57.4.17 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est )` [inherited]

Profile likelihood bounds computations with VM algorithm

Definition at line 220 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::MLEData::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBoundsData::profile_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb`, `mappel::estimator::ProfileBoundsData::profile_points_lb_rllh`, `mappel::estimator::ProfileBoundsData::profile_points_ub`, `mappel::estimator::ProfileBoundsData::profile_ub`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`.

9.57.4.18 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug ( const ModelDataT< Model > & data, ProfileBoundsDebugData & bounds_est ) [inherited]`

Profile likelihood bounds computations with VM algorithm

Definition at line 258 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsDebugData::estimated_idx`, `mappel::estimator::ProfileBoundsDebugData::mle`, `mappel::print_text_image()`, `mappel::estimator::Estimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::ProfileBoundsDebugData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::debug::error_bounds_profile_likelihood_debug()`.

9.57.4.19 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel ( const ModelDataT< Model > & data, ProfileBoundsData & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 464 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ProfileBoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::MLEData::obs`, `mappel::print_text_image()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::Estimator< Model >::record_walltime()`, `mappel::estimator::subroutine::solve_profile_initial_step()`, `mappel::estimator::ProfileBoundsData::target_rllh_delta`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`.

9.57.4.20 `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack ( const ModelDataStackT< Model > & data_stack, ProfileBoundsDataStack & bounds_est ) [override], [virtual], [inherited]`

Profile likelihood bounds computations with VM algorithm

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 500 of file estimator\_impl.h.

References `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::Error`, `mappel::estimator::ProfileBoundsData::estimated_idx`s, `mappel::estimator::ProfileBoundsDataStack::estimated_idx`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, and `mappel::estimator::ProfileBoundsDataStack::initialize_arrays()`.

mappel::estimator::ProfileBoundsData::mle, mappel::estimator::ProfileBoundsDataStack::mle, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileBoundsDataStack::Ndata, mappel::estimator::ProfileBoundsDataStack::Nparams\_est, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::MLEData::obsI, mappel::estimator::MLEDataStack::obsI, mappel::print\_text\_image(), mappel::estimator::ProfileBoundsData::profile\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb, mappel::estimator::ProfileBoundsData::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_lb\_rllh, mappel::estimator::ProfileBoundsData::profile\_points\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub, mappel::estimator::ProfileBoundsData::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsDataStack::profile\_points\_ub\_rllh, mappel::estimator::ProfileBoundsData::profile\_ub, mappel::estimator::ProfileBoundsDataStack::profile\_ub, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), mappel::estimator::MLEData::rllh, mappel::estimator::MLEDataStack::rllh, mappel::estimator::subroutine::solve\_profile\_initial\_step(), mappel::estimator::ProfileBoundsData::target\_rllh\_delta, mappel::estimator::ProfileBoundsDataStack::target\_rllh\_delta, mappel::estimator::MLEData::theta, and mappel::estimator::MLEDataStack::theta.

Referenced by mappel::methods::openmp::error\_bounds\_profile\_likelihood\_stack().

**9.57.4.21** `template<class Model> double mappel::estimator::Estimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const IdxVecT & fixed_idx, const ParamT< Model > & fixed_theta_init, StencilT< Model > & theta_max ) [inherited]`

Profile likelihood estimation methods

Definition at line 190 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::print\_text\_image(), mappel::estimator::Estimator< Model >::record\_exit\_code(), mappel::estimator::Estimator< Model >::record\_walltime(), and mappel::methods::objective::rllh().

**9.57.4.22** `template<class Model> void mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max ( const ModelDataT< Model > & data, const ParamVecT< Model > & fixed_theta_init, ProfileLikelihoodData & profile ) [override],[virtual],[inherited]`

Profile likelihood estimation methods

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 418 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::compute\_profile\_estimate(), mappel::estimator::Error, mappel::estimator::ProfileLikelihoodData::fixed\_idx, mappel::estimator::ProfileLikelihoodData::fixed\_values, mappel::estimator::Estimator< Model >::model, mappel::estimator::ProfileLikelihoodData::Nfixed, mappel::estimator::ThreadedEstimator< Model >::num\_threads, mappel::estimator::ProfileLikelihoodData::Nvalues, mappel::print\_text\_image(), mappel::estimator::ProfileLikelihoodData::profile\_likelihood, mappel::estimator::ProfileLikelihoodData::profile\_parameters, mappel::estimator::ThreadedEstimator< Model >::record\_exit\_code(), and mappel::estimator::Estimator< Model >::record\_walltime().

**9.57.4.23** `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_debug_stats ( )`  
`[virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 832 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx`s.

**9.57.4.24** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts ( ) const`  
`[inline], [inherited]`

Run statistics.

Definition at line 274 of file `estimator.h`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.57.4.25** `template<class Model > const Model & mappel::estimator::Estimator< Model >::get_model ( )`  
`[inherited]`

Definition at line 108 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::model`.

**9.57.4.26** `template<class Model > StatsT mappel::estimator::IterativeMaximizer< Model >::get_stats ( )`  
`[virtual], [inherited]`

Run statistics.

Reimplemented from [mappel::estimator::ThreadedEstimator< Model >](#).

Definition at line 811 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio`, `mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::max_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::Estimator< Model >::num_estimations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::methods::error_bounds_profile_likelihood()`, `mappel::methods::debug::error_bounds_profile_likelihood_debug()`, `mappel::methods::openmp::error_bounds_profile_likelihood_parallel()`, `mappel::methods::openmp::error_bounds_profile_likelihood_stack()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`.

**9.57.4.27** `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_backtracks ( ) const`  
`[inline], [inherited]`

Definition at line 441 of file estimator.h.

**9.57.4.28** `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals ( ) const`  
`[inline], [inherited]`

Definition at line 443 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.57.4.29** `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals ( ) const`  
`[inline], [inherited]`

Definition at line 442 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.57.4.30** `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_iterations ( ) const`  
`[inline], [inherited]`

Definition at line 440 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.57.4.31** `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::local_maximize ( const`  
`ModelDataT< Model> & im, StencilT< Model> & stencil, MLEData & data ) [inherited]`

Perform a local maximization to finish off a simulated annealing run.

Definition at line 1145 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.57.4.32** `template<class Model> void mappel::estimator::IterativeMaximizer< Model >::local_maximize ( const`  
`ModelDataT< Model> & im, StencilT< Model> & stencil, MLEDebugData & debug_data ) [inherited]`

Definition at line 1158 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model>::maximize()`, `mappel::estimator::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::estimator::MLEDebugData::obsI`, `mappel::estimator::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::stencil()`, `mappel::estimator::MLEDebugData::theta`, and `mappel::estimator::IterativeMaximizer< Model>::MaximizerData::theta()`.

9.57.4.33 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize ( const ModelDataT< Model > & im, const IdxVecT & fixed_param_idx, StencilT< Model > & stencil, MLEDebugData & mle ) [inherited]`

Definition at line 1173 of file estimator\_impl.h.

References `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::subroutine::bound_step()`, `mappel::clamp()`, `mappel::estimator::subroutine::compute_bound_scaling_vec()`, `mappel::estimator::subroutine::compute_cauchy_point()`, `mappel::estimator::subroutine::compute_initial_trust_radius()`, `mappel::estimator::subroutine::compute_quadratic_model_value()`, `mappel::estimator::subroutine::compute_scaled_problem()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::fixed_idx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::free_idx`, `mappel::estimator::Estimator< Model >::get_exit_counts()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals()`, `mappel::estimator::IterativeMaximizer< Model >::get_total_iterations()`, `mappel::methods::objective::grad()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::methods::objective::grad2()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters()`, `mappel::methods::objective::hessian()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::is_positive_definite()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::max_iterations`, `mappel::estimator::IterativeMaximizer< Model >::maximize()`, `mappel::estimator::MaxIter`, `mappel::estimator::IterativeMaximizer< Model >::min_profile_bound_residual`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters()`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obs`, `mappel::estimator::MLEDebugData::obs`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::MLEDebugData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s0`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::s1`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::estimator::MLEDebugData::sequence`, `mappel::estimator::MLEDebugData::sequence_rllh`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::estimator::subroutine::solve_TR_subproblem()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step`, `mappel::estimator::MLEData::theta`, `mappel::estimator::MLEDebugData::theta`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta()`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_iterations`, and `mappel::estimator::TrustRegionRadius`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.57.4.34 `template<class Model > virtual void mappel::estimator::IterativeMaximizer< Model >::maximize ( MaximizerData & data ) [protected],[pure virtual],[inherited]`

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

9.57.4.35 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_backtracks ( )`  
[inherited]

9.57.4.36 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_der_evals ( )`  
[inherited]

9.57.4.37 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_fun_evals ( )`  
[inherited]

9.57.4.38 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_iterations ( )`  
[inherited]

9.57.4.39 `template<class Model> std::string mappel::estimator::TrustRegionMaximizer< Model >::name ( ) const`  
[inline],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 630 of file estimator.h.

9.57.4.40 `template<class Model > bool mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack (`  
`MaximizerData & data, IdxT fixed_idx, double target_rllh, double old_fval, const VecT & fgrad )` [protected],  
[inherited]

Definition at line 943 of file estimator\_impl.h.

References [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_max\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_linear\\_step\\_ratio](#), [mappel::estimator::IterativeMaximizer< Model >::backtrack\\_min\\_ratio](#), [mappel::clamp\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::convergence\\_min\\_step\\_size\\_ratio](#), [mappel::methods::objective::grad\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::im](#), [mappel::estimator::IterativeMaximizer< Model >::max\\_backtracks](#), [mappel::estimator::MaxBacktracks](#), [mappel::estimator::Estimator< Model >::model](#), [mappel::norm\\_sq\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_backtrack\(\)](#), [mappel::estimator::ThreadedEstimator< Model >::record\\_exit\\_code\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::record\\_iteration\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::restore\\_stencil\(\)](#), [mappel::methods::objective::rllh\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::save\\_stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::saved\\_theta\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::set\\_stencil\(\)](#), [mappel::square\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::step](#), [mappel::estimator::StepSize](#), and [mappel::estimator::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local\\_profile\\_maximize\(\)](#).

**9.57.4.41** `template<class Model > void mappel::estimator::ThreadedEstimator< Model >::record_exit_code ( ExitCode code )` `[override], [protected], [virtual], [inherited]`

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 578 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::exit_counts`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::methods::observed_information()`, `mappel::estimator::MLEData::obsI`, `mappel::estimator::MLEData::rllh`, `mappel::methods::objective::rllh()`, `mappel::estimator::Success`, and `mappel::estimator::MLEData::theta`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.57.4.42** `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::record_run_statistics ( const MaximizerData & data )` `[protected], [inherited]`

Definition at line 859 of file `estimator_impl.h`.

References `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idxes()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idxes`, `mappel::estimator::ThreadedEstimator< Model >::mtx`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nBacktracks`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nIterations`, `mappel::estimator::IterativeMaximizer< Model >::total_backtracks`, `mappel::estimator::IterativeMaximizer< Model >::total_der_evals`, `mappel::estimator::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::estimator::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.57.4.43** `template<class Model > void mappel::estimator::Estimator< Model >::record_walltime ( ClockT::time_point start_walltime, int num_estimations )` `[protected], [inherited]`

Definition at line 360 of file `estimator_impl.h`.

References `mappel::estimator::Estimator< Model >::num_estimations`, and `mappel::estimator::Estimator< Model >::total_walltime`.

Referenced by `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds_debug()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`.



9.57.4.44 `template<class Model > void mappel::estimator::IterativeMaximizer< Model >::solve_profile_bound ( MaximizerData & data, MLEData & mle, double llh_delta, IdxT fixed_idx, IdxT which_bound )` [protected], [virtual], [inherited]

Definition at line 1137 of file estimator\_impl.h.

References mappel::estimator::Estimator< Model >::name().

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound(), and mappel::estimator::IterativeMaximizer< Model >::compute\_profile\_bound\_debug().

### 9.57.5 Member Data Documentation

9.57.5.1 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_max_ratio = 0.50` [static], [protected], [inherited]

Definition at line 460 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.57.5.2 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio = 1e-3` [static], [protected], [inherited]

Definition at line 461 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::backtrack(), and mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.57.5.3 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_ratio = 0.05` [static], [protected], [inherited]

Definition at line 459 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile\_bound\_backtrack().

9.57.5.4 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_function_change_ratio = 1.0e-9` [static], [protected], [inherited]

Convergence criteria: tolerance for function-value change.

Definition at line 456 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::convergence\_test\_grad\_ratio(), and mappel::estimator::IterativeMaximizer< Model >::get\_stats().

**9.57.5.5** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::convergence_min_step_size_ratio = 1.0e-9` [static], [protected], [inherited]

Convergence criteria: tolerance of relative step size.

Definition at line 457 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::convergence_test_step_size()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.57.5.6** `template<class Model> const double mappel::estimator::TrustRegionMaximizer< Model >::convergence_min_trust_radius = 1.0e-8` [static]

Convergence criteria: Minimum trust region radius.

Definition at line 625 of file estimator.h.

**9.57.5.7** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::DefaultIterations = 100` [static], [inherited]

Definition at line 428 of file estimator.h.

**9.57.5.8** `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 299 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.57.5.9** `template<class Model > IdxVecT mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx`s [protected], [inherited]

Debugging: Stores last set of `backtrack_idx`s when `data.save_seq==true`.

Definition at line 475 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.57.5.10** `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::max_backtracks = 8` [static], [protected], [inherited]

Definition at line 462 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`.

**9.57.5.11** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::max_iterations`  
`[protected], [inherited]`

Definition at line 466 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.57.5.12** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::max_threads`  
`[protected], [inherited]`

Definition at line 324 of file estimator.h.

**9.57.5.13** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model`  
`>::min_eigenvalue_correction_delta = 1e-3 [static], [protected], [inherited]`

Ensure the minimum eigenvalue is at least this big when correcting indefinite matrix.

Definition at line 454 of file estimator.h.

**9.57.5.14** `template<class Model > const double mappel::estimator::IterativeMaximizer< Model`  
`>::min_profile_bound_residual = 1e-4 [static], [protected], [inherited]`

Minimum residual in quadratic solutions of equation (8) to accept. Revert to newton step.

Definition at line 464 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`.

**9.57.5.15** `template<class Model > const Model& mappel::estimator::Estimator< Model >::model [protected],`  
`[inherited]`

Definition at line 294 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::backtrack()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate()`, `mappel::estimator::Estimator< Model >::compute_estimate_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_bound()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound()`, `mappel::estimator::Estimator< Model >::compute_profile_bound_debug()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound_debug()`, `mappel::estimator::Estimator< Model >::compute_profile_estimate()`, `mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::estimator::Estimator< Model >::estimate_max()`, `mappel::estimator::Estimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, `mappel::estimator::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::estimator::CGaussMLE< Model >::get_debug_stats()`, `mappel::estimator::Estimator< Model >::get_model()`, `mappel::estimator::IterativeMaximizer< Model >::local_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack()`, and `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`.

**9.57.5.16** `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` [protected], [inherited]

Definition at line 326 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, `mappel::estimator::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.57.5.17** `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [protected], [inherited]

Definition at line 297 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get_stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, `mappel::estimator::HeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussHeuristicEstimator< Model >::get_stats()`, `mappel::estimator::CGaussMLE< Model >::get_stats()`, `mappel::estimator::SimulatedAnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >::record_walltime()`.

**9.57.5.18** `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 325 of file estimator.h.

Referenced by `mappel::estimator::ThreadedEstimator< Model >::clear_stats()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`, `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_max()`, and `mappel::estimator::ThreadedEstimator< Model >::get_stats()`.

**9.57.5.19** `template<class Model> const double mappel::estimator::TrustRegionMaximizer< Model >::rho_cauchy_min = 0.1` [static]

Minimum acceptable rho for cauchy point: Coleman beta / Bellavia beta\_1.

Definition at line 619 of file estimator.h.

**9.57.5.20** `template<class Model> const double mappel::estimator::TrustRegionMaximizer< Model >::rho_obj_min = 0.25` [static]

Minimum acceptable rho: Coleman mu / Bellavia beta\_2.

Definition at line 620 of file estimator.h.

**9.57.5.21** `template<class Model> const double mappel::estimator::TrustRegionMaximizer< Model >::rho_obj_opt = 0.75`  
`[static]`

Optimal step rho: Coleman eta / Bellavia beta\_2.

Definition at line 621 of file estimator.h.

**9.57.5.22** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_backtracks = 0`  
`[protected], [inherited]`

Definition at line 470 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.57.5.23** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_der_evals = 0`  
`[protected], [inherited]`

Definition at line 472 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::Simulated↵AnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel↵::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.57.5.24** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_fun_evals = 0`  
`[protected], [inherited]`

Definition at line 471 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::Simulated↵AnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel↵::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.57.5.25** `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_iterations = 0`  
`[protected], [inherited]`

Definition at line 469 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::clear_stats()`, `mappel::estimator::Simulated↵AnnealingMaximizer< Model >::get_stats()`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, `mappel↵::estimator::IterativeMaximizer< Model >::local_profile_maximize()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

**9.57.5.26** `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected], [inherited]`

Definition at line 298 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::clear_stats()`, `mappel::estimator::Estimator< Model >::get↵stats()`, `mappel::estimator::ThreadedEstimator< Model >::get_stats()`, and `mappel::estimator::Estimator< Model >↵::record_walltime()`.

9.57.5.27 `template<class Model> const double mappel::estimator::TrustRegionMaximizer< Model  
>::trust_radius_decrease = 0.25 [static]`

Trust radius decrease ratio to step size: Coleman gamma\_1 / Bellavia alpha\_2.

Definition at line 623 of file estimator.h.

9.57.5.28 `template<class Model> const double mappel::estimator::TrustRegionMaximizer< Model  
>::trust_radius_decrease_min = 0.125 [static]`

Smallest allowable trust radius decrease ratio: Coleman gamma\_0 / Bellavia alpha\_1.

Definition at line 622 of file estimator.h.

9.57.5.29 `template<class Model> const double mappel::estimator::TrustRegionMaximizer< Model  
>::trust_radius_increase = 2 [static]`

Trust radius increase ratio: Coleman gamma\_2 / Bellavia alpha\_3.

Definition at line 624 of file estimator.h.

The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator\\_impl.h](#)

## 10 File Documentation

### 10.1 display.cpp File Reference

```
#include "Mappel/display.h"
```

#### Namespaces

- [mappel](#)

#### Functions

- `const char * mappel::lambda\_term\_color (int size, int Lidx)`
- `ostream & mappel::print\_centered\_title (ostream &out, char fill, int width, const char *title=nullptr)`
- `ostream & mappel::print\_labeled\_image (ostream &out, const arma::mat &im, const char *title, const char *color)`
- `template<>  
std::ostream & mappel::print\_image (std::ostream &out, const arma::vec &im)`
- `template<>  
std::ostream & mappel::print\_image (std::ostream &out, const arma::mat &im)`
- `template<>  
std::ostream & mappel::print\_text\_image (std::ostream &out, const arma::vec &im)`
- `template<>  
std::ostream & mappel::print\_text\_image (std::ostream &out, const arma::mat &im)`
- `template<>  
std::ostream & mappel::print\_image (std::ostream &out, const arma::cube &im)`

### Variables

- const char \* [mappel::TERM\\_BLACK](#) ="1;30"
- const char \* [mappel::TERM\\_RED](#) ="1;31"
- const char \* [mappel::TERM\\_GREEN](#) ="1;32"
- const char \* [mappel::TERM\\_YELLOW](#) ="1;33"
- const char \* [mappel::TERM\\_BLUE](#) ="1;34"
- const char \* [mappel::TERM\\_MAGENTA](#) ="1;35"
- const char \* [mappel::TERM\\_CYAN](#) ="1;36"
- const char \* [mappel::TERM\\_WHITE](#) ="1;37"
- const char \* [mappel::TERM\\_DIM\\_BLACK](#) ="0;30"
- const char \* [mappel::TERM\\_DIM\\_RED](#) ="0;31"
- const char \* [mappel::TERM\\_DIM\\_GREEN](#) ="0;32"
- const char \* [mappel::TERM\\_DIM\\_YELLOW](#) ="0;33"
- const char \* [mappel::TERM\\_DIM\\_BLUE](#) ="0;34"
- const char \* [mappel::TERM\\_DIM\\_MAGENTA](#) ="0;35"
- const char \* [mappel::TERM\\_DIM\\_CYAN](#) ="0;36"
- const char \* [mappel::TERM\\_DIM\\_WHITE](#) ="0;37"

#### 10.1.1 Detailed Description

##### Author

Mark J. Olah ([mjo@cs.unm.edu](mailto:mjo@cs.unm.edu))

##### Date

2014-2019

## 10.2 display.h File Reference

Textual image display with colors.

```
#include <iostream>
#include <iomanip>
#include <armadillo>
```

### Namespaces

- [mappel](#)

### Functions

- template<class ImageT >  
std::ostream & [mappel::print\\_image](#) (std::ostream &out, const ImageT &im)
- template<class ImageT >  
std::ostream & [mappel::print\\_text\\_image](#) (std::ostream &out, const ImageT &im)
- template<class Vec >  
std::ostream & [mappel::print\\_vec\\_row](#) (std::ostream &out, const Vec &vec, const char \*header, int header\_width, const char \*color=nullptr)

### 10.2.1 Detailed Description

Textual image display with colors.

#### Author

Mark J. Olah ([mjo@cs.unm.edu](mailto:mjo@cs.unm.edu))

#### Date

2014-2019

## 10.3 estimator.cpp File Reference

Non-templated estimator helper routines and static constants.

```
#include <cmath>
#include <iomanip>
#include <armadillo>
#include "Mappel/util.h"
#include "Mappel/numerical.h"
#include "Mappel/estimator.h"
#include "Mappel/estimator_helpers.h"
```

#### Namespaces

- [mappel](#)
- [mappel::estimator](#)
- [mappel::estimator::subroutine](#)

#### Functions

- [VecT mappel::estimator::subroutine::solve\\_profile\\_initial\\_step](#) (const MatT &obsI, IdxT fixed\_idx, double llh\_delta)
- [VecT mappel::estimator::subroutine::bound\\_step](#) (const VecT &step, const VecT &theta, const VecT &lbound, const VecT &ubound)
- [void mappel::estimator::subroutine::compute\\_bound\\_scaling\\_vec](#) (const VecT &theta, const VecT &g, const VecT &lbound, const VecT &ubound, VecT &v, VecT &Jv)
- [VecT mappel::estimator::subroutine::compute\\_D\\_scale](#) (const VecT &oldDscale, const VecT &grad2)
- [void mappel::estimator::subroutine::compute\\_scaled\\_problem](#) (const MatT &H, const VecT &g, const VecT &Dinv, const VecT &Jv, MatT &Hhat, VecT &ghat)
- [double mappel::estimator::subroutine::compute\\_initial\\_trust\\_radius](#) (const VecT &ghat)
- [VecT mappel::estimator::subroutine::compute\\_cauchy\\_point](#) (const VecT &g, const MatT &H, double delta)
- [double mappel::estimator::subroutine::compute\\_quadratic\\_model\\_value](#) (const VecT &s, const VecT &g, const MatT &H)  
*Quadratic model value at given step Compute a quadratic model.*
- [VecT mappel::estimator::subroutine::solve\\_TR\\_subproblem](#) (const VecT &g, const MatT &H, double delta)  
*Exact solver the TR sub-problem even for non-positive definite H.*
- [VecT mappel::estimator::subroutine::solve\\_restricted\\_step\\_length\\_newton](#) (const VecT &g, const MatT &H, double delta, double lambda\_lb, double lambda\_ub)



### 10.3.1 Detailed Description

Non-templated estimator helper routines and static constants.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.4 estimator.h File Reference

The class declaration and inline and templated functions for the Estimator class hierarchy.

```
#include <exception>
#include <fstream>
#include <string>
#include <limits>
#include <memory>
#include <mutex>
#include <map>
#include "Mappel/rng.h"
#include "cGaussMLE/cGaussMLE.h"
#include <chrono>
#include "Mappel/util.h"
```

#### Classes

- struct [mappel::estimator::MLEData](#)
- struct [mappel::estimator::MLEDebugData](#)
- struct [mappel::estimator::MLEDataStack](#)
- struct [mappel::estimator::ProfileLikelihoodData](#)
- struct [mappel::estimator::ProfileBoundsData](#)
- struct [mappel::estimator::ProfileBoundsDebugData](#)
- struct [mappel::estimator::ProfileBoundsDataStack](#)
- class [mappel::estimator::Estimator< Model >](#)
- class [mappel::estimator::ThreadedEstimator< Model >](#)
- class [mappel::estimator::HeuristicEstimator< Model >](#)
- class [mappel::estimator::CGaussHeuristicEstimator< Model >](#)
- class [mappel::estimator::CGaussMLE< Model >](#)
- class [mappel::estimator::SimulatedAnnealingMaximizer< Model >](#)
- class [mappel::estimator::IterativeMaximizer< Model >](#)
- class [mappel::estimator::IterativeMaximizer< Model >::MaximizerData](#)
- class [mappel::estimator::NewtonDiagonalMaximizer< Model >](#)
- class [mappel::estimator::NewtonMaximizer< Model >](#)
- class [mappel::estimator::QuasiNewtonMaximizer< Model >](#)
- class [mappel::estimator::TrustRegionMaximizer< Model >](#)

## Namespaces

- [mappel](#)
- [mappel::estimator](#)

## Typedefs

- typedef `std::chrono::high_resolution_clock` [ClockT](#)
- enum [mappel::estimator::ExitCode](#) : `IdxT` {  
[mappel::estimator::ExitCode::TrustRegionRadius](#) = 9, [mappel::estimator::ExitCode::ModelImprovement](#) = 8,  
[mappel::estimator::ExitCode::GradRatio](#) = 7, [mappel::estimator::ExitCode::FunctionValue](#) = 6,  
[mappel::estimator::ExitCode::StepSize](#) = 5, [mappel::estimator::ExitCode::Success](#) = 4, [mappel::estimator::ExitCode::MaxBacktracks](#) = 3, [mappel::estimator::ExitCode::MaxIter](#) = 2,  
[mappel::estimator::ExitCode::Unassigned](#) = 1, [mappel::estimator::ExitCode::Error](#) = 0 }

### 10.4.1 Detailed Description

The class declaration and inline and templated functions for the Estimator class hierarchy.

## Author

Mark J. Olah ([mjo@cs.unm DOT edu](mailto:mjo@cs.unm DOT edu))

## Date

2014-2019

### 10.4.2 Class Documentation

#### 10.4.2.1 struct `mappel::estimator::MLEData`

Data reporting structures A maximum-likelihood estimate for a single image. A container to group the necessary information at an MLEstimate

Definition at line 40 of file `estimator.h`.

## Class Members

<a href="#">MatT</a>	<code>obsI</code>	Observed Fisher information matrix at theta.
<code>double</code>	<code>rllh</code>	RLLH at theta.
<a href="#">VecT</a>	<code>theta</code>	Theta estimate.

## 10.4.2.2 struct mappel::estimator::MLEDebugData

A maximum-likelihood estimate for a single image with debugging information. A container to group the necessary information at an MLEstimate

Definition at line 50 of file estimator.h.

## Class Members

<a href="#">IdxT</a>	Nseq	Number of points evaluated including theta_init and theta_mle.
<a href="#">MatT</a>	obsI	Observed Fisher information matrix at theta.
double	rllh	RLLH at theta.
<a href="#">MatT</a>	sequence	Sequence of evaluated points including theta_init and theta_mle.
<a href="#">VecT</a>	sequence_rllh	RLLH at each point in sequence.
<a href="#">VecT</a>	theta	Theta estimate.

## 10.4.2.3 struct mappel::estimator::MLEDataStack

A stack of maximum-likelihood estimates for a stack of images A container to group the necessary information at an MLEstimate

Definition at line 63 of file estimator.h.

## Class Members

<a href="#">IdxT</a>	Ndata	Number of data estimates.
<a href="#">CubeT</a>	obsI	Observed Fisher information matrix stack. size:[Nparams,Nparams,Ndata].
<a href="#">VecT</a>	rllh	RLLH stack. size:[Ndata].
<a href="#">MatT</a>	theta	Theta estimate stack. size:[Nparams,Ndata].

## 10.4.2.4 struct mappel::estimator::ProfileLikelihoodData

Container for profile likelihood estimator data Includes both controlling (input) parameters as well as reporting (ouput) parameters to give output parameters context.

Definition at line 74 of file estimator.h.

## Class Members

<a href="#">IdxVecT</a>	fixed_idx	Indexes of fixed parameters.
<a href="#">MatT</a>	fixed_values	Vector values for each fixed parameter size:[Nfixed,Nvalues];.
<a href="#">IdxT</a>	Nfixed	Number of fixed parameters.
<a href="#">IdxT</a>	Nvalues	Number of values of fixed parameters evaluated.
<a href="#">VecT</a>	profile_likelihood	profile likelihood for each column of fixed parameter values
<a href="#">MatT</a>	profile_parameters	Points at which the profile likelihood maximum was obtained.

#### 10.4.2.5 struct mappel::estimator::ProfileBoundsDebugData

Data for debugging of estimation of profile bounds for a single parameter of a single image Includes both controlling (input) parameters as well as reporting (ouptut) parameters to give output parameters context.

Definition at line 113 of file estimator.h.

##### Class Members

<a href="#">IdxT</a>	estimated_idx	Index of single parameter to estimate for.
<a href="#">MLEData</a>	mle	Theta maximum-likelihood estimate, rllh, and Obsl.
<a href="#">IdxT</a>	Nseq_lb	Number of points in sequence_lb.
<a href="#">IdxT</a>	Nseq_ub	Number of points in sequence_ub.
double	profile_lb	size:[Nparams_est] Lower bound estimated for estimated_idx.
double	profile_ub	size:[Nparams_est] Upper bound estimated for estimated_idx.
<a href="#">MatT</a>	sequence_lb	size:[NumParams,Nseq_lb] Sequence of evaluated points for lb estimate (including theta mle as initial point)
<a href="#">VecT</a>	sequence_lb_rllh	size:[Nseq_lb] RLLH at each of the sequence_lb points
<a href="#">MatT</a>	sequence_ub	size:[NumParams,Nseq_ub] Sequence of evaluated points for ub estimate (including theta mle as initial point)
<a href="#">VecT</a>	sequence_ub_rllh	size:[Nseq_ub] RLLH at each of the sequence_ub points
double	target_rllh_delta	Targeted rllh change in value from MLE ( $-\chi^2_{inv}(\text{confidence},1)/2$ )

#### 10.4.3 Typedef Documentation

##### 10.4.3.1 typedef std::chrono::high\_resolution\_clock ClockT

Definition at line 25 of file estimator.h.

## 10.5 estimator\_helpers.h File Reference

Estimator helper subroutines.

##### Namespaces

- [mappel](#)
- [mappel::estimator](#)
- [mappel::estimator::subroutine](#)

## Functions

- VecT [mappel::estimator::subroutine::bound\\_step](#) (const VecT &step, const VecT &theta, const VecT &lbound, const VecT &ubound)
- void [mappel::estimator::subroutine::compute\\_bound\\_scaling\\_vec](#) (const VecT &theta, const VecT &g, const VecT &lbound, const VecT &ubound, VecT &v, VecT &Jv)
- VecT [mappel::estimator::subroutine::compute\\_D\\_scale](#) (const VecT &oldDscale, const VecT &grad2)
- void [mappel::estimator::subroutine::compute\\_scaled\\_problem](#) (const MatT &H, const VecT &g, const VecT &Dinv, const VecT &Jv, MatT &Hhat, VecT &ghat)
- VecT [mappel::estimator::subroutine::solve\\_profile\\_initial\\_step](#) (const MatT &obsI, IdxT fixed\_idx, double llh\_delta)
- double [mappel::estimator::subroutine::compute\\_initial\\_trust\\_radius](#) (const VecT &ghat)
- VecT [mappel::estimator::subroutine::compute\\_cauchy\\_point](#) (const VecT &g, const MatT &H, double delta)
- double [mappel::estimator::subroutine::compute\\_quadratic\\_model\\_value](#) (const VecT &s, const VecT &g, const MatT &H)  
*Quadratic model value at given step Compute a quadratic model.*
- VecT [mappel::estimator::subroutine::solve\\_TR\\_subproblem](#) (const VecT &g, const MatT &H, double delta)  
*Exact solver the TR sub-problem even for non-positive definite H.*
- VecT [mappel::estimator::subroutine::solve\\_restricted\\_step\\_length\\_newton](#) (const VecT &g, const MatT &H, double delta, double lambda\_lb, double lambda\_ub)

## 10.5.1 Detailed Description

Estimator helper subroutines.

## Author

Mark J. Olah (mjo@cs.unm DOT edu)

## Date

2014-2019

## 10.6 estimator\_impl.h File Reference

```
#include <thread>
#include <cmath>
#include <armadillo>
#include "Mappel/estimator.h"
#include "Mappel/estimator_helpers.h"
#include "Mappel/rng.h"
#include "Mappel/numerical.h"
#include "Mappel/display.h"
```

## Namespaces

- [mappel](#)
- [mappel::estimator](#)

## Functions

- `template<class Model >`  
`std::ostream & mappel::estimator::operator<< (std::ostream &out, Estimator< Model > &estimator)`

### 10.6.1 Detailed Description

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.7 estimator\_statics.cpp File Reference

```
#include "Mappel/Gauss1DMAP.h"  
#include "Mappel/estimator_impl.h"
```

#### Namespaces

- [mappel](#)

### 10.7.1 Detailed Description

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

01-15-2014

## 10.8 Gauss1DMAP.cpp File Reference

The class definition and template Specializations for Gauss1DMAP.

```
#include "Mappel/Gauss1DMAP.h"
```

#### Namespaces

- [mappel](#)

### 10.8.1 Detailed Description

The class definition and template Specializations for Gauss1DMAP.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.9 Gauss1DMAP.h File Reference

The class declaration and inline and templated functions for Gauss1DMAP.

```
#include "Mappel/Gauss1DModel.h"
#include "Mappel/PoissonNoise1DObjective.h"
#include "Mappel/MAPEstimator.h"
#include "Mappel/model_methods.h"
```

#### Classes

- class [mappel::Gauss1DMAP](#)  
*A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.*

#### Namespaces

- [mappel](#)

### 10.9.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DMAP.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.10 Gauss1DMLE.cpp File Reference

The class definition and template Specializations for Gauss1DMLE.

```
#include "Mappel/Gauss1DMLE.h"
```

### Namespaces

- [mappel](#)

### 10.10.1 Detailed Description

The class definition and template Specializations for Gauss1DMLE.

### Author

Mark J. Olah (mjo@cs.unm DOT edu)

### Date

2014-2019

## 10.11 Gauss1DMLE.h File Reference

The class declaration and inline and templated functions for Gauss1DMLE.

```
#include "Mappel/Gauss1DModel.h"  
#include "Mappel/PoissonNoise1DObjective.h"  
#include "Mappel/MLEstimator.h"  
#include "Mappel/model_methods.h"
```

### Classes

- class [mappel::Gauss1DMLE](#)  
*A 1D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective.*

### Namespaces

- [mappel](#)



### 10.11.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DMLE.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.12 Gauss1DModel.cpp File Reference

The class definition and template Specializations for Gauss1DModel.

```
#include "Mappel/Gauss1DModel.h"  
#include "Mappel/stencil.h"
```

#### Namespaces

- [mappel](#)

#### Functions

- `std::ostream & mappel::operator<< (std::ostream &out, const Gauss1DModel::Stencil &s)`

### 10.12.1 Detailed Description

The class definition and template Specializations for Gauss1DModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.13 Gauss1DModel.h File Reference

The class declaration and inline and templated functions for Gauss1DModel.

```
#include "Mappel/PointEmitterModel.h"  
#include "Mappel/ImageFormat1DBase.h"  
#include "Mappel/MCMCA adaptor1D.h"
```

## Classes

- class [mappel::Gauss1DModel](#)  
*A base class for 1D Gaussian PSF with a fixed sigma (standard dev.)*
- class [mappel::Gauss1DModel::Stencil](#)  
*Stencil for 1D fixed-sigma models.*

## Namespaces

- [mappel](#)

### 10.13.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

### 10.14 Gauss1DsMAP.cpp File Reference

The class definition and template Specializations for Gauss1DsMAP.

```
#include "Mappel/Gauss1DsMAP.h"
```

## Namespaces

- [mappel](#)

### 10.14.1 Detailed Description

The class definition and template Specializations for Gauss1DsMAP.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2017

## 10.15 Gauss1DsMAP.h File Reference

The class declaration and inline and templated functions for Gauss1DsMAP.

```
#include "Mappel/Gauss1DsModel.h"  
#include "Mappel/PoissonNoise1DObjective.h"  
#include "Mappel/MAPEstimator.h"  
#include "Mappel/model_methods.h"
```

### Classes

- class [mappel::Gauss1DsMAP](#)  
*A 1D Gaussian with variable PSF sigma under an Poisson read noise assumption and MAP Objective.*

### Namespaces

- [mappel](#)

#### 10.15.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DsMAP.

### Author

Mark J. Olah (mjo@cs.unm DOT edu)

### Date

2014-2019

## 10.16 Gauss1DsMLE.cpp File Reference

The class definition and template Specializations for Gauss1DsMLE.

```
#include "Mappel/Gauss1DsMLE.h"
```

### Namespaces

- [mappel](#)

### 10.16.1 Detailed Description

The class definition and template Specializations for Gauss1DsMLE.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2017

### 10.17 Gauss1DsMLE.h File Reference

The class declaration and inline and templated functions for Gauss1DsMLE.

```
#include "Mappel/Gauss1DsModel.h"
#include "Mappel/PoissonNoise1DObjective.h"
#include "Mappel/MLEstimator.h"
#include "Mappel/model_methods.h"
```

#### Classes

- class [mappel::Gauss1DsMLE](#)  
*A 1D Gaussian with variable PSF under an Poisson noise assumption and maximum-likelihood estimator.*

#### Namespaces

- [mappel](#)

### 10.17.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DsMLE.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.18 Gauss1DsModel.cpp File Reference

The class definition and template Specializations for Gauss1DsModel.

```
#include "Mappel/Gauss1DsModel.h"
#include "Mappel/stencil.h"
```

### Namespaces

- [mappel](#)

### Functions

- `std::ostream & mappel::operator<< (std::ostream &out, const Gauss1DsModel::Stencil &s)`

#### 10.18.1 Detailed Description

The class definition and template Specializations for Gauss1DsModel.

### Author

Mark J. Olah (mjo@cs.unm DOT edu)

### Date

2014-2019

## 10.19 Gauss1DsModel.h File Reference

The class declaration and inline and templated functions for Gauss1DsModel.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/ImageFormat1DBase.h"
#include "Mappel/MCMCA adaptor1Ds.h"
```

### Classes

- class [mappel::Gauss1DsModel](#)  
*Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels.*
- class [mappel::Gauss1DsModel::Stencil](#)  
*Stencil for 1D variable-sigma models.*

## Namespaces

- [mappel](#)

### 10.19.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DsModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

### 10.20 Gauss2DMAP.cpp File Reference

The class definition and template Specializations for Gauss2DMAP.

```
#include "Mappel/Gauss2DMAP.h"
```

## Namespaces

- [mappel](#)

### 10.20.1 Detailed Description

The class definition and template Specializations for Gauss2DMAP.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

### 10.21 Gauss2DMAP.h File Reference

The class declaration and inline and templated functions for Gauss2DMAP.

```
#include "Mappel/Gauss2DModel.h"  
#include "Mappel/PoissonNoise2DObjective.h"  
#include "Mappel/MAPEstimator.h"  
#include "Mappel/model_methods.h"
```

## Classes

- class [mappel::Gauss2DMP](#)  
*A 2D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.*

## Namespaces

- [mappel](#)

### 10.21.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DMP.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.22 Gauss2DMLE.cpp File Reference

The class definition and template Specializations for Gauss2DMLE.

```
#include "Mappel/Gauss2DMLE.h"
```

## Namespaces

- [mappel](#)

### 10.22.1 Detailed Description

The class definition and template Specializations for Gauss2DMLE.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.23 Gauss2DMLE.h File Reference

The class declaration and inline and templated functions for Gauss2DMLE.

```
#include "Mappel/Gauss2DModel.h"
#include "Mappel/PoissonNoise2DObjective.h"
#include "Mappel/MLEstimator.h"
#include "Mappel/model_methods.h"
```

### Classes

- class [mappel::Gauss2DMLE](#)  
*A 2D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective.*

### Namespaces

- [mappel](#)

### 10.23.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DMLE.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.24 Gauss2DModel.cpp File Reference

The class definition and template Specializations for Gauss2DModel.

```
#include "Mappel/Gauss2DModel.h"
#include "Mappel/stencil.h"
```

### Namespaces

- [mappel](#)



## Functions

- `std::ostream & mappel::operator<<` (`std::ostream &out`, `const Gauss2DModel::Stencil &s`)

### 10.24.1 Detailed Description

The class definition and template Specializations for Gauss2DModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.25 Gauss2DModel.h File Reference

The class declaration and inline and templated functions for Gauss2DModel.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/ImageFormat2DBase.h"
#include "Mappel/MCMCApaptor2D.h"
#include "Mappel/Gauss1DMAP.h"
```

## Classes

- class `mappel::Gauss2DModel`  
*A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma.*
- class `mappel::Gauss2DModel::Stencil`  
*Stencil for 2D fixed-sigma models.*

## Namespaces

- `mappel`

## Functions

- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DModel, Model >::value, ParamT< Model > >::type mappel::cgauss←`  
`_heuristic_compute_estimate` (`const Model &model`, `const ModelDataT< Model > &im`, `const ParamT< Model`  
`> &theta_init`)
- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DModel, Model >::value, ParamT< Model > >::type mappel::cgauss←`  
`_compute_estimate` (`Model &model`, `const ModelDataT< Model > &im`, `const ParamT< Model > &theta_init`, `int`  
`max_iterations`)
- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DModel, Model >::value, ParamT< Model > >::type mappel::cgauss←`  
`_compute_estimate_debug` (`const Model &model`, `const ModelDataT< Model > &im`, `const ParamT< Model >`  
`&theta_init`, `int max_iterations`, `ParamVecT< Model > &sequence`)

### 10.25.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.26 Gauss2DsMAP.cpp File Reference

The class definition and template Specializations for Gauss2DsMAP.

```
#include "Mappel/Gauss2DsMAP.h"
```

#### Namespaces

- [mappel](#)

### 10.26.1 Detailed Description

The class definition and template Specializations for Gauss2DsMAP.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.27 Gauss2DsMAP.h File Reference

The class declaration and inline and templated functions for Gauss2DsMAP.

```
#include "Mappel/Gauss2DsModel.h"  
#include "Mappel/PoissonNoise2DObjective.h"  
#include "Mappel/MLEstimator.h"  
#include "Mappel/model_methods.h"
```

## Classes

- class [mappel::Gauss2DsMAP](#)

*A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum a-posteriori objective.*

## Namespaces

- [mappel](#)

### 10.27.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsMAP.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.28 Gauss2DsMLE.cpp File Reference

The class definition and template Specializations for Gauss2DsMLE.

```
#include "Mappel/Gauss2DsMLE.h"
```

## Namespaces

- [mappel](#)

### 10.28.1 Detailed Description

The class definition and template Specializations for Gauss2DsMLE.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.29 Gauss2DsMLE.h File Reference

The class declaration and inline and templated functions for Gauss2DsMLE.

```
#include "Mappel/Gauss2DsModel.h"
#include "Mappel/PoissonNoise2DObjective.h"
#include "Mappel/MLEstimator.h"
#include "Mappel/model_methods.h"
```

### Classes

- class [mappel::Gauss2DsMLE](#)  
*A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum-likelihood objective.*

### Namespaces

- [mappel](#)

### 10.29.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsMLE.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.30 Gauss2DsModel.cpp File Reference

The class definition and template Specializations for Gauss2DsModel.

```
#include "Mappel/Gauss2DsModel.h"
#include "Mappel/stencil.h"
```

### Namespaces

- [mappel](#)

## Functions

- `std::ostream & mappel::operator<<` (`std::ostream &out`, `const Gauss2DsModel::Stencil &s`)

### 10.30.1 Detailed Description

The class definition and template Specializations for Gauss2DsModel.

## Author

Mark J. Olah (mjo@cs.unm DOT edu)

## Date

2014-2019

## 10.31 Gauss2DsModel.h File Reference

The class declaration and inline and templated functions for Gauss2DsModel.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/ImageFormat2DBase.h"
#include "Mappel/MCMCApaptor2Ds.h"
#include "Mappel/Gauss1DsMAP.h"
```

## Classes

- class `mappel::Gauss2DsModel`  
*A base class for 2D Gaussian PSF where the gaussian sigma is controlled by a single scalar parameter which is called `sigma_ratio`. The size of the gaussian psf is `sigma_ratio*psf_sigma`, where `psf_sigma` is considered as a vector [`psf_sigmaX`, `psf_sigmaY`].*
- class `mappel::Gauss2DsModel::Stencil`  
*Stencil for 2D scalar-sigma models.*

## Namespaces

- `mappel`

## Functions

- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DsModel, Model >::value, ParamT< Model > >::type mappel←`  
`::cgauss_heuristic_compute_estimate` (const Model &model, const ModelDataT< Model > &im, const ParamT< Model > &theta\_init)
- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DsModel, Model >::value, ParamT< Model > >::type mappel←`  
`::cgauss_compute_estimate` (Model &model, const ModelDataT< Model > &im, const ParamT< Model > &theta\_init, int max\_iterations)
- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DsModel, Model >::value, ParamT< Model > >::type mappel←`  
`::cgauss_compute_estimate_debug` (const Model &model, const ModelDataT< Model > &im, const ParamT< Model > &theta\_init, int max\_iterations, ParamVecT< Model > &sequence)

### 10.31.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

### 10.32 Gauss2DsxymAP.h File Reference

The class declaration and inline and templated functions for Gauss2DsxymAP.

```
#include "Mappel/PoissonNoise2DObjective.h"
#include "Mappel/MAPEstimator.h"
#include "Mappel/Gauss1DModel.h"
#include "Mappel/model_methods.h"
```

## Classes

- class `mappel::Gauss2DsxymAP`  
*A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.*

## Namespaces

- `mappel`

## 10.32.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsxyMAP.

## Author

Mark J. Olah (mjo@cs.unm DOT edu)

## Date

2017

## 10.33 Gauss2DsxyModel.h File Reference

The class declaration and inline and templated functions for Gauss2DsxyModel.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/ImageFormat2DBase.h"
#include "Mappel/Gauss1DsMAP.h"
```

## Classes

- class [mappel::Gauss2DsxyModel](#)  
A base class for 2D Gaussian PSF with axis-aligned gaussian with free parameters for both  $\sigma_x$  and  $\sigma_y$ . Gaussian sigma parameters  $\sigma_x$  and  $\sigma_y$  are measured in units of pixels. The model has 6 parameters,  $[x, y, l, b, \sigma_x, \sigma_y]$ .
- class [mappel::Gauss2DsxyModel::Stencil](#)  
*Stencil* for 2D free-sigma (astigmatic) models.

## Namespaces

- [mappel](#)

## Functions

- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DsxyModel, Model >::value, ParamT< Model > >::type` [mappel::cgau](#)  
`ss_heuristic_compute_estimate` (const Model &model, const ModelDataT< Model > &im, const ParamT< Model > &theta\_init)
- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DsxyModel, Model >::value, ParamT< Model > >::type` [mappel::cgau](#)  
`ss_compute_estimate` (Model &model, const ModelDataT< Model > &im, const ParamT< Model > &theta\_init, int max\_iterations)
- `template<class Model >`  
`std::enable_if< std::is_base_of< Gauss2DsxyModel, Model >::value, ParamT< Model > >::type` [mappel::cgau](#)  
`ss_compute_estimate_debug` (const Model &model, const ModelDataT< Model > &im, const ParamT< Model > &theta\_init, int max\_iterations, ParamVecT< Model > &sequence)

### 10.33.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DskyModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.34 ImageFormat1DBase.cpp File Reference

The class definition and template Specializations for ImageFormat1DBase.

```
#include "Mappel/ImageFormat1DBase.h"
```

#### Namespaces

- [mappel](#)

### 10.34.1 Detailed Description

The class definition and template Specializations for ImageFormat1DBase.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.35 ImageFormat1DBase.h File Reference

The class declaration and inline and templated functions for ImageFormat1DBase.

```
#include "Mappel/util.h"  
#include "Mappel/ImageFormat2DBase.h"
```



## Classes

- class [mappel::ImageFormat1DBase](#)  
*A virtual base class for 2D image localization objectives.*

## Namespaces

- [mappel](#)
- [mappel::methods](#)  
*Templated functions for operating on a [PointEmitterModel](#).*

## Functions

- `template<class Model >`  
`ReturnIfSubclassT< ImageT< Model >, Model, ImageFormat1DBase > mappel::methods::model\_image (const`  
`Model &model, const StencilT< Model > &s)`

### 10.35.1 Detailed Description

The class declaration and inline and templated functions for ImageFormat1DBase.

## Author

Mark J. Olah ([mjo@cs.unm DOT edu](mailto:mjo@cs.unm DOT edu))

## Date

2014-2019 The virtual base class for all point 2D image based emitter Models and Objectives

## 10.36 ImageFormat2DBase.cpp File Reference

The class definition and template Specializations for ImageFormat2DBase.

```
#include "Mappel/ImageFormat2DBase.h"
```

## Namespaces

- [mappel](#)

### 10.36.1 Detailed Description

The class definition and template Specializations for ImageFormat2DBase.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.37 ImageFormat2DBase.h File Reference

The class declaration and inline and templated functions for ImageFormat2DBase.

```
#include "Mappel/util.h"
```

#### Classes

- class [mappel::ImageFormat2DBase](#)  
*A virtual base class for 2D image localization objectives.*

#### Namespaces

- [mappel](#)
- [mappel::methods](#)  
*Templated functions for operating on a [PointEmitterModel](#).*

#### Functions

- `template<class Model >`  
`ReturnIfSubclassT< ImageT< Model >, Model, ImageFormat2DBase > mappel::methods::model\_image (const Model &model, const typename Model::Stencil &s)`

### 10.37.1 Detailed Description

The class declaration and inline and templated functions for ImageFormat2DBase.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019 The virtual base class for all point 2D image based emitter Models and Objectives

## 10.38 Install.md File Reference

### 10.39 MAPEstimator.h File Reference

Class declaration and inline and templated functions for MAPEstimator.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/MLEstimator.h"
```

#### Classes

- class [mappel::MAPEstimator](#)  
*A Mixin class to configure a for MLE estimation (null prior).*

#### Namespaces

- [mappel](#)
- [mappel::methods](#)  
*Templated functions for operating on a [PointEmitterModel](#).*
- [mappel::methods::objective](#)
- [mappel::methods::objective::debug](#)

#### Functions

- `template<class Model >`  
`ReturnIfSubclassT< double, Model, MAPEstimator > mappel::methods::objective::llh (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< double, Model, MAPEstimator > mappel::methods::objective::rllh (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< ParamT< Model >, Model, MAPEstimator > mappel::methods::objective::grad (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, MAPEstimator > mappel::methods::objective::grad2 (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad, ParamT< Model > &grad2)`
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, MAPEstimator > mappel::methods::objective::hessian (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad, MatT &hess)`
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, MAPEstimator > mappel::methods::objective::debug::llh\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, MAPEstimator > mappel::methods::objective::debug::rllh\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, MAPEstimator > mappel::methods::objective::debug::grad\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< CubeT, Model, MAPEstimator > mappel::methods::objective::debug::hessian\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`

### 10.39.1 Detailed Description

Class declaration and inline and templated functions for MAPEstimator.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2017-2019

## 10.40 mcmc.cpp File Reference

MCMC helper functions.

```
#include "Mappel/util.h"
#include "Mappel/mcmc_data.h"
```

#### Namespaces

- [mappel](#)
- [mappel::mcmc](#)

#### Functions

- `IdxT mappel::mcmc::num\_oversample (IdxT Nsample, IdxT Nburnin, IdxT thin)`
- `MatT mappel::mcmc::thin\_sample (MatT &sample, IdxT Nburnin, IdxT thin)`
- `void mappel::mcmc::thin\_sample (const MatT &sample, const VecT &sample_rllh, IdxT Nburnin, IdxT thin, MatT &subsample, VecT &subsample_rllh)`

### 10.40.1 Detailed Description

MCMC helper functions.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.41 mcmc.h File Reference

Templated MCMC methods for posterior estimation.

```
#include <cmath>
#include "Mappel/mcmc_data.h"
#include "Mappel/util.h"
#include "Mappel/rng.h"
#include <trng/uniform01_dist.hpp>
```

### Namespaces

- [mappel](#)
- [mappel::mcmc](#)

### Functions

- [IdxT mappel::mcmc::num\\_oversample](#) (IdxT Nsample, IdxT Nburnin, IdxT thin)
- [MatT mappel::mcmc::thin\\_sample](#) (MatT &sample, IdxT Nburnin, IdxT thin)
- [void mappel::mcmc::thin\\_sample](#) (const MatT &sample, const VecT &sample\_rllh, IdxT Nburnin, IdxT thin, MatT &subsample, VecT &subsample\_rllh)
- [void mappel::mcmc::estimate\\_sample\\_posterior](#) (const MatT &sample, VecT &theta\_posterior\_mean, MatT &theta\_posterior\_cov)
- [template<class Mat, class Vec>](#)  
[void mappel::mcmc::compute\\_posterior\\_credible](#) (const Mat &sample, double confidence, Vec &lb, Vec &ub)
- [template<class Model>](#)  
[void mappel::mcmc::sample\\_posterior](#) (const Model &model, const ModelDataT< Model > &im, const StencilT< Model > &theta\_init, MatT &sample, VecT &sample\_rllh)
- [template<class Model>](#)  
[void mappel::mcmc::sample\\_posterior\\_debug](#) (const Model &model, const ModelDataT< Model > &im, const StencilT< Model > &theta\_init, MatT &sample, VecT &sample\_rllh, MatT &candidate, VecT &candidate\_rllh)

### 10.41.1 Detailed Description

Templated MCMC methods for posterior estimation.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2015-2019

## 10.42 mcmc\_data.h File Reference

MCMC data storage types.

```
#include <armadillo>
```

### Classes

- struct [mappel::mcmc::MCMCData](#)
- struct [mappel::mcmc::MCMCDebugData](#)
- struct [mappel::mcmc::MCMCDataStack](#)

### Namespaces

- [mappel](#)
- [mappel::mcmc](#)

### 10.42.1 Detailed Description

MCMC data storage types.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2015-2019

## 10.43 MCMCAdaptor1D.cpp File Reference

The class definition and template Specializations for MCMCAdaptor1D.

```
#include "Mappel/MCMCAdaptor1D.h"
```

### Namespaces

- [mappel](#)

### 10.43.1 Detailed Description

The class definition and template Specializations for MCMCAdaptor1D.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2018

## 10.44 MCMCAdaptor1D.h File Reference

The class declaration and inline and templated functions for MCMCAdaptor1D.

```
#include "Mappel/MCMCAdaptorBase.h"  
#include "Mappel/PointEmitterModel.h"
```

#### Classes

- class [mappel::MCMCAdaptor1D](#)

#### Namespaces

- [mappel](#)

### 10.44.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptor1D.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2018-2019

## 10.45 MCMCAdaptor1Ds.cpp File Reference

The class definition and template Specializations for MCMCAdaptor1Ds.

```
#include "Mappel/MCMCAdaptor1Ds.h"
```

## Namespaces

- [mappel](#)

### 10.45.1 Detailed Description

The class definition and template Specializations for MCMCAdaptor1Ds.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2018

### 10.46 MCMCAdaptor1Ds.h File Reference

The class declaration and inline and templated functions for MCMCAdaptor1Ds.

```
#include "Mappel/MCMCAdaptor1D.h"
```

## Classes

- class [mappel::MCMCAdaptor1Ds](#)

## Namespaces

- [mappel](#)

### 10.46.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptor1Ds.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2018-2019



## 10.47 MCMCAdaptor2D.cpp File Reference

The class definition and template Specializations for MCMCAdaptor2D.

```
#include "Mappel/MCMCAdaptor2D.h"
```

### Namespaces

- [mappel](#)

#### 10.47.1 Detailed Description

The class definition and template Specializations for MCMCAdaptor2D.

### Author

Mark J. Olah (mjo@cs.unm DOT edu)

### Date

2018

## 10.48 MCMCAdaptor2D.h File Reference

The class declaration and inline and templated functions for MCMCAdaptor2D.

```
#include "Mappel/MCMCAdaptor1D.h"
```

### Classes

- class [mappel::MCMCAdaptor2D](#)

### Namespaces

- [mappel](#)

#### 10.48.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptor2D.

### Author

Mark J. Olah (mjo@cs.unm DOT edu)

### Date

2018-2019

## 10.49 MCMCAdaptor2Ds.cpp File Reference

The class definition and template Specializations for MCMCAdaptor2Ds.

```
#include "Mappel/MCMCAdaptor2Ds.h"
```

### Namespaces

- [mappel](#)

### 10.49.1 Detailed Description

The class definition and template Specializations for MCMCAdaptor2Ds.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2018

## 10.50 MCMCAdaptor2Ds.h File Reference

The class declaration and inline and templated functions for MCMCAdaptor2Ds.

```
#include "Mappel/MCMCAdaptor2D.h"
```

### Classes

- class [mappel::MCMCAdaptor2Ds](#)

### Namespaces

- [mappel](#)

### 10.50.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptor2Ds.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2018-2019

## 10.51 MCMCAdaptorBase.cpp File Reference

The class definition and template Specializations for MCMCAdaptorBase.

```
#include "Mappel/MCMCAdaptorBase.h"
```

### Namespaces

- [mappel](#)

#### 10.51.1 Detailed Description

The class definition and template Specializations for MCMCAdaptorBase.

### Author

Mark J. Olah (mjo@cs.unm DOT edu)

### Date

2014-2018

## 10.52 MCMCAdaptorBase.h File Reference

The class declaration and inline and templated functions for MCMCAdaptorBase.

```
#include "Mappel/util.h"
```

### Classes

- class [mappel::MCMCAdaptorBase](#)

### Namespaces

- [mappel](#)

#### 10.52.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptorBase.

### Author

Mark J. Olah (mjo@cs.unm DOT edu)

### Date

2018

## 10.53 MLEstimator.h File Reference

Class declaration and inline and templated functions for MLEstimator.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/MAPEstimator.h"
```

### Classes

- class [mappel::MLEstimator](#)  
*A Mixin class to configure a for MLE estimation (null prior).*

### Namespaces

- [mappel](#)
- [mappel::methods](#)  
*Templated functions for operating on a [PointEmitterModel](#).*
- [mappel::methods::objective](#)
- [mappel::methods::objective::debug](#)

### Functions

- `template<class Model >`  
`ReturnIfSubclassT< double, Model, MLEstimator > mappel::methods::objective::llh (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< double, Model, MLEstimator > mappel::methods::objective::rllh (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< ParamT< Model >, Model, MLEstimator > mappel::methods::objective::grad (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, MLEstimator > mappel::methods::objective::grad2 (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad, ParamT< Model > &grad2)`
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, MLEstimator > mappel::methods::objective::hessian (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad, MatT &hess)`
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, MLEstimator > mappel::methods::objective::debug::llh\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, MLEstimator > mappel::methods::objective::debug::rllh\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, MLEstimator > mappel::methods::objective::debug::grad\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< CubeT, Model, MLEstimator > mappel::methods::objective::debug::hessian\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`

## 10.53.1 Detailed Description

Class declaration and inline and templated functions for MLEstimator.

## Author

Mark J. Olah (mjo@cs.unm DOT edu)

## Date

2017

10.54 `model_methods.h` File Reference

```
#include "Mappel/mcmc_data.h"
#include "Mappel/estimator.h"
#include "Mappel/mcmc.h"
#include "Mappel/openmp_methods.h"
#include "Mappel/model_methods_impl.h"
#include "Mappel/estimator_impl.h"
```

## Namespaces

- [mappel](#)
- [mappel::methods](#)
  - Templated functions for operating on a *PointEmitterModel*.
- [mappel::methods::objective](#)
- [mappel::methods::objective::debug](#)
- [mappel::methods::debug](#)

## Functions

- `template<class Model >`  
`ImageT< Model > mappel::methods::model\_image (const Model &model, const ParamT< Model > &theta)`
- `template<class Model, class rng_t >`  
`ModelDataT< Model > mappel::methods::simulate\_image (const Model &model, const ParamT< Model > &theta)`
- `template<class Model, class rng_t >`  
`ModelDataT< Model > mappel::methods::simulate\_image (const Model &model, const ParamT< Model > &theta, rng_t &rng)`
- `template<class Model >`  
`ModelDataT< Model > mappel::methods::simulate\_image (const Model &model, const StencilT< Model > &s)`
- `template<class Model >`  
`ModelDataT< Model > mappel::methods::simulate\_image\_from\_model (const Model &model, const ImageT< Model > &model_im)`

- `template<class Model >`  
`double mappel::methods::objective::llh (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`double mappel::methods::objective::rllh (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`ParamT< Model > mappel::methods::objective::grad (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`ParamT< Model > mappel::methods::objective::grad2 (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`void mappel::methods::objective::grad2 (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, ParamT< Model > &grad_val, ParamT< Model > &grad2_val)`
- `template<class Model >`  
`MatT mappel::methods::objective::hessian (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`MatT mappel::methods::objective::hessian (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`void mappel::methods::objective::hessian (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, ParamT< Model > &grad, MatT &hess)`
- `template<class Model >`  
`void mappel::methods::objective::hessian (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, MatT &hess)`
- `template<class Model >`  
`MatT mappel::methods::objective::negative_definite_hessian (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`MatT mappel::methods::objective::negative_definite_hessian (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`void mappel::methods::objective::negative_definite_hessian (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, ParamT< Model > &grad, MatT &hess)`
- `template<class Model >`  
`void mappel::methods::objective::negative_definite_hessian (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad, MatT &hess)`
- `template<class Model >`  
`VecT mappel::methods::objective::debug::llh_components (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`VecT mappel::methods::objective::debug::rllh_components (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`MatT mappel::methods::objective::debug::grad_components (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`CubeT mappel::methods::objective::debug::hessian_components (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta)`
- `template<class Model >`  
`void mappel::methods::aposteriori_objective (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, double &rllh, ParamT< Model > &grad, MatT &hess)`

- template<class Model >  
void [mappel::methods::aposteriori\\_objective](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- template<class Model >  
void [mappel::methods::prior\\_objective](#) (const Model &model, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- template<class Model >  
void [mappel::methods::likelihood\\_objective](#) (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s, double &rllh, ParamT< Model > &grad, MatT &hess)
- template<class Model >  
void [mappel::methods::likelihood\\_objective](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- template<class Model >  
ParamT< Model > [mappel::methods::cr\\_lower\\_bound](#) (const Model &model, const typename Model::Stencil &s)  
*Calculate the Cramer-Rao lower bound at the given parameters.*
- template<class Model >  
ParamT< Model > [mappel::methods::cr\\_lower\\_bound](#) (const Model &model, const ParamT< Model > &theta)
- template<class Model >  
MatT [mappel::methods::expected\\_information](#) (const Model &model, const ParamT< Model > &theta)
- template<class Model >  
MatT [mappel::methods::observed\\_information](#) (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta\_mle)
- template<class Model >  
MatT [mappel::methods::observed\\_information](#) (const Model &model, const ModelDataT< Model > &data, const StencilT< Model > &theta\_mle)
- template<class Model >  
void [mappel::methods::estimate\\_max](#) (const Model &model, const ModelDataT< Model > &data, const std::string &method, estimator::MLEData &mle)
- template<class Model >  
void [mappel::methods::estimate\\_max](#) (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta\_init, estimator::MLEData &mle)
- template<class Model >  
void [mappel::methods::estimate\\_max](#) (const Model &model, const ModelDataT< Model > &data, const std::string &method, estimator::MLEData &mle, StatsT &stats)
- template<class Model >  
void [mappel::methods::estimate\\_max](#) (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta\_init, estimator::MLEData &mle, StatsT &stats)
- template<class Model >  
double [mappel::methods::estimate\\_profile\\_likelihood](#) (const Model &model, const ModelDataT< Model > &data, const std::string &method, const IdxVecT &fixed\_idx, const ParamT< Model > &fixed\_theta\_init)
- template<class Model >  
double [mappel::methods::estimate\\_profile\\_likelihood](#) (const Model &model, const ModelDataT< Model > &data, const std::string &method, const IdxVecT &fixed\_idx, const ParamT< Model > &fixed\_theta\_init, StencilT< Model > &profile\_max)
- template<class Model >  
double [mappel::methods::estimate\\_profile\\_likelihood](#) (const Model &model, const ModelDataT< Model > &data, const std::string &method, const IdxVecT &fixed\_idx, const ParamT< Model > &fixed\_theta\_init, StencilT< Model > &profile\_max, StatsT &stats)
- template<class Model >  
void [mappel::methods::estimate\\_profile\\_likelihood](#) (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta\_init, estimator::ProfileLikelihoodData &profile\_data)

- `template<class Model >`  
`void mappel::methods::estimate_posterior (const Model &model, const ModelDataT< Model > &data, mcmc::MCMCData &mcmc_est)`
- `template<class Model >`  
`void mappel::methods::estimate_posterior (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, mcmc::MCMCData &mcmc_est)`
- `template<class Model >`  
`void mappel::methods::error_bounds_expected (const Model &model, const ParamT< Model > &theta_est, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)`
- `template<class Model >`  
`void mappel::methods::error_bounds_observed (const Model &model, const estimator::MLEData &mle, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)`
- `template<class Model >`  
`void mappel::methods::error_bounds_profile_likelihood (const Model &model, const ModelDataT< Model > &data, estimator::ProfileBoundsData &bounds)`
- `template<class Model >`  
`void mappel::methods::error_bounds_profile_likelihood (const Model &model, const ModelDataT< Model > &data, estimator::ProfileBoundsData &bounds, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::error_bounds_posterior_credible (const Model &model, const MatT &sample, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)`
- `template<class Model >`  
`void mappel::methods::debug::estimate_max_debug (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, estimator::MLEDebugData &mle, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::debug::error_bounds_profile_likelihood_debug (const Model &model, const ModelDataT< Model > &data, estimator::ProfileBoundsDebugData &bounds, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::debug::estimate_posterior_debug (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, mcmc::MCMCDebugData &mcmc_debug_sample)`

## 10.55 model\_methods\_impl.h File Reference

```
#include "Mappel/numerical.h"
```

### Namespaces

- [mappel](#)
- [mappel::methods](#)
  - Templated functions for operating on a [PointEmitterModel](#).*
- [mappel::methods::objective](#)
- [mappel::methods::objective::debug](#)
- [mappel::methods::debug](#)



## Functions

- template<class Model >  
Model::ImageT [mappel::methods::model\\_image](#) (const Model &model, const ParamT< Model > &theta)
- template<class Model >  
ModelDataT< Model > [mappel::methods::simulate\\_image](#) (const Model &model, const ParamT< Model > &theta)
- template<class Model , class RngT >  
ModelDataT< Model > [mappel::methods::simulate\\_image](#) (const Model &model, const ParamT< Model > &theta, RngT &rng)
- template<class Model >  
ModelDataT< Model > [mappel::methods::simulate\\_image](#) (const Model &model, const StencilT< Model > &s)
- template<class Model >  
ModelDataT< Model > [mappel::methods::simulate\\_image\\_from\\_model](#) (const Model &model, const ImageT< Model > &model\_im)
- template<class Model >  
double [mappel::methods::objective::llh](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- template<class Model >  
double [mappel::methods::objective::rllh](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- template<class Model >  
ParamT< Model > [mappel::methods::objective::grad](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- template<class Model >  
ParamT< Model > [mappel::methods::objective::grad2](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- template<class Model >  
void [mappel::methods::objective::grad2](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta, ParamT< Model > &grad\_val, ParamT< Model > &grad2\_val)
- template<class Model >  
MatT [mappel::methods::objective::hessian](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- template<class Model >  
MatT [mappel::methods::objective::hessian](#) (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- template<class Model >  
void [mappel::methods::objective::hessian](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta, ParamT< Model > &grad, MatT &hess)
- template<class Model >  
void [mappel::methods::objective::hessian](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta, MatT &hess)
- template<class Model >  
MatT [mappel::methods::objective::negative\\_definite\\_hessian](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- template<class Model >  
MatT [mappel::methods::objective::negative\\_definite\\_hessian](#) (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s)
- template<class Model >  
void [mappel::methods::objective::negative\\_definite\\_hessian](#) (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta, ParamT< Model > &grad, MatT &hess)
- template<class Model >  
void [mappel::methods::objective::negative\\_definite\\_hessian](#) (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s, ParamT< Model > &grad, MatT &hess)

- `template<class Model >`  
`VecT mappel::methods::objective::debug::llh_components` (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- `template<class Model >`  
`VecT mappel::methods::objective::debug::rllh_components` (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- `template<class Model >`  
`MatT mappel::methods::objective::debug::grad_components` (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- `template<class Model >`  
`CubeT mappel::methods::objective::debug::hessian_components` (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta)
- `template<class Model >`  
`void mappel::methods::aposteriori_objective` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`  
`void mappel::methods::prior_objective` (const Model &model, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`  
`void mappel::methods::likelihood_objective` (const Model &model, const ModelDataT< Model > &data\_im, const StencilT< Model > &s, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`  
`void mappel::methods::aposteriori_objective` (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`  
`void mappel::methods::likelihood_objective` (const Model &model, const ModelDataT< Model > &data\_im, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`  
`ParamT< Model > mappel::methods::cr_lower_bound` (const Model &model, const typename Model::Stencil &s)  
*Calculate the Cramer-Rao lower bound at the given parameters.*
- `template<class Model >`  
`ParamT< Model > mappel::methods::cr_lower_bound` (const Model &model, const ParamT< Model > &theta)
- `template<class Model >`  
`MatT mappel::methods::expected_information` (const Model &model, const ParamT< Model > &theta)
- `template<class Model >`  
`MatT mappel::methods::observed_information` (const Model &model, const ModelDataT< Model > &data, const StencilT< Model > &theta\_mle)
- `template<class Model >`  
`MatT mappel::methods::observed_information` (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta\_mle)
- `template<class Model >`  
`void mappel::methods::estimate_max` (const Model &model, const ModelDataT< Model > &data, const std::string &method, estimator::MLEData &mle)
- `template<class Model >`  
`void mappel::methods::estimate_max` (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta\_init, estimator::MLEData &mle)
- `template<class Model >`  
`void mappel::methods::estimate_max` (const Model &model, const ModelDataT< Model > &data, const std::string &method, estimator::MLEData &mle, StatsT &stats)
- `template<class Model >`  
`void mappel::methods::estimate_max` (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta\_init, estimator::MLEData &mle, StatsT &stats)

- `template<class Model >`  
`double mappel::methods::estimate_profile_likelihood (const Model &model, const ModelDataT< Model > &data, const std::string &method, const IdxVecT &fixed_idx, const ParamT< Model > &fixed_theta_init)`
- `template<class Model >`  
`double mappel::methods::estimate_profile_likelihood (const Model &model, const ModelDataT< Model > &data, const std::string &method, const IdxVecT &fixed_idx, const ParamT< Model > &fixed_theta_init, StencilT< Model > &profile_max)`
- `template<class Model >`  
`double mappel::methods::estimate_profile_likelihood (const Model &model, const ModelDataT< Model > &data, const std::string &method, const IdxVecT &fixed_idx, const ParamT< Model > &fixed_theta_init, StencilT< Model > &profile_max, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::estimate_posterior (const Model &model, const ModelDataT< Model > &data, mcmc::MCMCData &mcmc_est)`
- `template<class Model >`  
`void mappel::methods::estimate_posterior (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, mcmc::MCMCData &mcmc_est)`
- `template<class Model >`  
`void mappel::methods::error_bounds_expected (const Model &model, const ParamT< Model > &theta_est, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)`
- `template<class Model >`  
`void mappel::methods::error_bounds_observed (const Model &model, const estimator::MLEData &mle, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)`
- `template<class Model >`  
`void mappel::methods::error_bounds_profile_likelihood (const Model &model, const ModelDataT< Model > &data, estimator::ProfileBoundsData &bounds)`
- `template<class Model >`  
`void mappel::methods::error_bounds_profile_likelihood (const Model &model, const ModelDataT< Model > &data, estimator::ProfileBoundsData &bounds, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::error_bounds_posterior_credible (const Model &model, const MatT &sample, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)`
- `template<class Model >`  
`void mappel::methods::debug::estimate_max_debug (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, estimator::MLEDebugData &mle, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::debug::error_bounds_profile_likelihood_debug (const Model &model, const ModelDataT< Model > &data, estimator::ProfileBoundsDebugData &bounds, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::debug::estimate_posterior_debug (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, mcmc::MCMCDebugData &mcmc_debug_sample)`

## 10.56 numerical.cpp File Reference

Numerical matrix operations.

```
#include <cassert>
#include "Mappel/numerical.h"
```

## Namespaces

- [mappel](#)

## Functions

- void [mappel::copy\\_Usym\\_mat](#) (arma::mat &usym)
- void [mappel::copy\\_Usym\\_mat\\_stack](#) (arma::cube &usym\_stack)
- void [mappel::copy\\_Lsym\\_mat](#) (arma::mat &lsym)
- void [mappel::cholesky\\_make\\_negative\\_definite](#) (arma::mat &m)
- void [mappel::cholesky\\_make\\_positive\\_definite](#) (arma::mat &m)
- bool [mappel::is\\_negative\\_definite](#) (const arma::mat &usym)
- bool [mappel::is\\_positive\\_definite](#) (const arma::mat &usym)
- bool [mappel::is\\_symmetric](#) (const arma::mat &A)
- void [mappel::cholesky\\_convert\\_lower\\_triangular](#) (arma::mat &chol)
- void [mappel::cholesky\\_convert\\_full\\_matrix](#) (arma::mat &chol)
- bool [mappel::cholesky](#) (arma::mat &A)
- bool [mappel::modified\\_cholesky](#) (arma::mat &A)
- arma::vec [mappel::cholesky\\_solve](#) (const arma::mat &C, const arma::vec &b)
- double [mappel::norm\\_sq](#) (const VecT &v)

### 10.56.1 Detailed Description

Numerical matrix operations.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2015-2019

### 10.57 numerical.h File Reference

Numerical matrix operations.

```
#include <cmath>
#include <climits>
#include <armadillo>
#include "Mappel/util.h"
```

## Namespaces

- [mappel](#)

## Functions

- void [mappel::copy\\_Usym\\_mat](#) (arma::mat &usym)
- void [mappel::copy\\_Usym\\_mat\\_stack](#) (arma::cube &usym\_stack)
- void [mappel::copy\\_Lsym\\_mat](#) (arma::mat &lsym)
- void [mappel::cholesky\\_convert\\_lower\\_triangular](#) (arma::mat &chol)
- void [mappel::cholesky\\_convert\\_full\\_matrix](#) (arma::mat &chol)
- void [mappel::cholesky\\_make\\_negative\\_definite](#) (arma::mat &m)
- void [mappel::cholesky\\_make\\_positive\\_definite](#) (arma::mat &m)
- bool [mappel::is\\_positive\\_definite](#) (const arma::mat &usym)
- bool [mappel::is\\_negative\\_definite](#) (const arma::mat &usym)
- bool [mappel::is\\_symmetric](#) (const arma::mat &A)
- bool [mappel::cholesky](#) (arma::mat &A)
- bool [mappel::modified\\_cholesky](#) (arma::mat &A)
- arma::vec [mappel::cholesky\\_solve](#) (const arma::mat &C, const arma::vec &b)
- template<class FloatT >  
FloatT [mappel::clamp](#) (FloatT val, FloatT min\_val, FloatT max\_val)
- double [mappel::norm\\_sq](#) (const VecT &v)

## 10.57.1 Detailed Description

Numerical matrix operations.

## Author

Mark J. Olah (mjo@cs.unm DOT edu)

## Date

2015-2019

## 10.58 OMPEXceptionCatcher.h File Reference

A lightweight class for managing C++ exception handling strategies for OpenMP methods.

```
#include <exception>
#include <mutex>
#include <functional>
#include <cstdint>
```

## Classes

- class [omp\\_exception\\_catcher::impl::OMPEXceptionCatcher<\\_dummy>](#)

## Namespaces

- [omp\\_exception\\_catcher](#)
- [omp\\_exception\\_catcher::impl\\_](#)

## Typedefs

- using [omp\\_exception\\_catcher::OMPEExceptionCatcher](#) = [impl\\_::OMPEExceptionCatcher](#)<>

## Enumerations

- enum [omp\\_exception\\_catcher::Strategy](#) { [omp\\_exception\\_catcher::Strategy::DoNotTry](#), [omp\\_exception\\_catcher::Strategy::Continue](#), [omp\\_exception\\_catcher::Strategy::Abort](#), [omp\\_exception\\_catcher::Strategy::RethrowFirst](#) }

### 10.58.1 Detailed Description

A lightweight class for managing C++ exception handling strategies for OpenMP methods.

#### Author

Mark J. Olah ([mjo@cs.unm.edu](mailto:mjo@cs.unm.edu))

#### Date

2019

#### Copyright

See LICENSE file OpenMP code must catch any exceptions that may have been thrown before exiting the OpenMP block. This class acts as lightweight wrapper that allows an arbitrary function or lambda expression to be run safely and efficiently in OMP even if it might throw exceptions. We employ one of 4 possible strategies as determined By the [omp\\_exception\\_catcher::Strategies](#) enum.

Strategy's : [omp\\_exception\\_catcher::Strategies::DoNotTry](#) – Don't even try, this is a null op to completely disable this class's effect. [omp\\_exception\\_catcher::Strategies::Continue](#) – Catch exceptions and keep going [omp\\_exception\\_catcher::Strategies::Abort](#) – Catch exceptions and abort [omp\\_exception\\_catcher::Strategies::RethrowFirst](#) – Re-throws first exception thrown by any thread.

Example usage: [omp\\_exception\\_catcher::OMPEExceptionCatcher](#) catcher([omp\\_exception\\_catcher::Strategies::Continue](#)); #pragma omp parallel for for(int n=0; n < N; n++) catcher.run([&]{ my\_output(n)=do\_my\_calculations(args(n)); }) catcher.rethrow(); //Required only if you ever might use RethrowFirst strategy

## 10.59 openmp\_methods.h File Reference

Namespaces for OpenMP parallelized versions of the mappel::model namespace functions (external methods)

```
#include <omp.h>
#include "Mappel/OMPExcceptionCatcher/OMPExcceptionCatcher.h"
#include "Mappel/util.h"
#include "Mappel/mcmc.h"
```

### Namespaces

- [mappel](#)
- [mappel::methods](#)
  - Templated functions for operating on a [PointEmitterModel](#).*
- [mappel::methods::openmp](#)
- [mappel::methods::objective](#)
- [mappel::methods::objective::openmp](#)

### Functions

- `template<class Model >`  
`void mappel::methods::openmp::sample\_prior\_stack (const Model &model, ParamVecT< Model > &theta_stack)`  
*Parallel sampling of the model prior.*
- `template<class Model >`  
`void mappel::methods::openmp::model\_image\_stack (const Model &model, const ParamVecT< Model > &theta_stack, ImageStackT< Model > &image_stack)`  
*Parallel computation of the model image.*
- `template<class Model >`  
`void mappel::methods::openmp::simulate\_image\_stack (const Model &model, const ParamVecT< Model > &theta_stack, ImageStackT< Model > &image_stack)`  
*Parallel simulation of images from one or more theta.*
- `template<class Model >`  
`void mappel::methods::openmp::cr\_lower\_bound\_stack (const Model &model, const ParamVecT< Model > &theta_stack, ParamVecT< Model > &crib_stack)`
- `template<class Model >`  
`void mappel::methods::openmp::expected\_information\_stack (const Model &model, const ParamVecT< Model > &theta_stack, CubeT &fisherI_stack)`
- `template<class Model >`  
`void mappel::methods::openmp::estimate\_max\_stack (const Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, estimator::MLEDataStack &mle_data_stack)`
- `template<class Model >`  
`void mappel::methods::openmp::estimate\_max\_stack (const Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_init_stack, estimator::MLEDataStack &mle_data_stack)`
- `template<class Model >`  
`void mappel::methods::openmp::estimate\_max\_stack (const Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_init_stack, estimator::MLEDataStack &mle_data_stack, StatsT &stats)`

- `template<class Model >`  
`void mappel::methods::openmp::estimate_profile_likelihood_stack (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamVecT< Model > &fixed_theta_init, estimator::ProfileLikelihoodData &est)`
- `template<class Model >`  
`void mappel::methods::openmp::estimate_profile_likelihood_stack (const Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamVecT< Model > &fixed_theta_init, estimator::ProfileLikelihoodData &est, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::openmp::estimate_posterior_stack (const Model &model, const ModelDataStackT< Model > &data_stack, const ParamVecT< Model > &theta_init_stack, mcmc::MCMCDataStack &est)`
- `template<class Model >`  
`void mappel::methods::openmp::estimate_posterior_stack (const Model &model, const ModelDataStackT< Model > &data_stack, mcmc::MCMCDataStack &est)`
- `template<class Model >`  
`void mappel::methods::openmp::error_bounds_expected_stack (const Model &model, const MatT &theta_est_stack, double confidence, MatT &theta_lb_stack, MatT &theta_ub_stack)`
- `template<class Model >`  
`void mappel::methods::openmp::error_bounds_observed_stack (const Model &model, const MatT &theta_est_stack, CubeT &obsl_stack, double confidence, MatT &theta_lb_stack, MatT &theta_ub_stack)`
- `template<class Model >`  
`void mappel::methods::openmp::error_bounds_profile_likelihood_parallel (const Model &model, const ModelDataStackT< Model > &image, estimator::ProfileBoundsData &est, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::openmp::error_bounds_profile_likelihood_parallel (const Model &model, const ModelDataT< Model > &image, estimator::ProfileBoundsData &est)`
- `template<class Model >`  
`void mappel::methods::openmp::error_bounds_profile_likelihood_stack (const Model &model, const ModelDataStackT< Model > &image, estimator::ProfileBoundsDataStack &est, StatsT &stats)`
- `template<class Model >`  
`void mappel::methods::openmp::error_bounds_profile_likelihood_stack (const Model &model, const ModelDataStackT< Model > &image, estimator::ProfileBoundsDataStack &est)`
- `template<class Model >`  
`void mappel::methods::objective::openmp::llh_stack (const Model &model, const ImageT< Model > &image, const ParamVecT< Model > &theta_stack, VecT &llh_stack)`  
*Parallel log\_likelihood calculations for a single image.*
- `template<class Model >`  
`void mappel::methods::objective::openmp::llh_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, VecT &llh_stack)`  
*Parallel log\_likelihood calculations for a stack of images.*
- `template<class Model >`  
`void mappel::methods::objective::openmp::rllh_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, VecT &rllh_stack)`  
*Parallel relative log\_likelihood calculations for a stack of images.*
- `template<class Model >`  
`void mappel::methods::objective::openmp::rllh_stack (const Model &model, const ImageT< Model > &image, const ParamVecT< Model > &theta_stack, VecT &rllh_stack)`
- `template<class Model >`  
`void mappel::methods::objective::openmp::grad_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, ParamVecT< Model > &grad_stack)`  
*Parallel model gradient calculations for a stack of images.*



- `template<class Model >`  
`void mappel::methods::objective::openmp::hessian_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, CubeT &hessian_stack)`  
*Parallel model Hessian calculations for a stack of images.*
- `template<class Model >`  
`void mappel::methods::objective::openmp::negative_definite_hessian_stack (const Model &model, const ImageStackT< Model > &image_stack, const ParamVecT< Model > &theta_stack, CubeT &hessian_stack)`  
*Parallel model negative\_definite Hessian approximation calculations for a stack of images.*

### 10.59.1 Detailed Description

Namespaces for OpenMP parallelized versions of the mappel::model namespace functions (external methods)

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2013-2019 OpenMP computation for stacked Model operations on vector data.

#### Design Decisions

- OpenMP vectorized versions are implemented as templated external methods in inline namespaces openmp. This allows easy future replacement with other parallelization mechanisms (CUDA, C++11 threads, etc..). Also allows the vectorized versions to directly overload with the non-vectorized base-versions.
- Because we want to integrate as seamlessly as possible with matlab, we use the armadillo package which stores arrays in column major order.
- Therefore in the \*\_stack operations, if they are to be parallelized, we want the data stored as a nParms X n matrix, i.e. each column is a parameter matrix. Similarly stacks are size X size X n, so that contiguous images sequences are contiguous in memory. This avoids false sharing.

## 10.60 PointEmitterModel.cpp File Reference

The class definition and template Specializations for PointEmitterModel.

```
#include <cmath>
#include <algorithm>
#include "Mappel/PointEmitterModel.h"
```

#### Namespaces

- [mappel](#)

### 10.60.1 Detailed Description

The class definition and template Specializations for PointEmitterModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

03-13-2014

### 10.61 PointEmitterModel.h File Reference

The class declaration and inline and templated functions for PointEmitterModel.

```
#include <iostream>
#include <string>
#include <armadillo>
#include <PriorHessian/CompositeDist.h>
#include <PriorHessian/TruncatedNormalDist.h>
#include <PriorHessian/ScaledSymmetricBetaDist.h>
#include <PriorHessian/TruncatedGammaDist.h>
#include <PriorHessian/TruncatedParetoDist.h>
#include "Mappel/util.h"
#include "Mappel/stencil.h"
#include "Mappel/display.h"
#include "Mappel/rng.h"
```

#### Classes

- class [mappel::PointEmitterModel](#)  
*A virtual Base type for point emitter localization models.*

#### Namespaces

- [mappel](#)

#### Functions

- template<class Model , typename = EnableIfSubclassT<Model,PointEmitterModel>>  
std::ostream & [mappel::operator<<](#) (std::ostream &out, const Model &model)

### 10.61.1 Detailed Description

The class declaration and inline and templated functions for PointEmitterModel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019 The base class for all point emitter localization models

## 10.62 PoissonGaussianNoise2DObjective.cpp File Reference

```
#include "Mappel/PoissonGaussianNoise2DObjective.h"
```

#### Namespaces

- [mappel](#)

## 10.63 PoissonGaussianNoise2DObjective.h File Reference

The class declaration and inline and templated functions for [PoissonGaussianNoise2DObjective](#).

```
#include "Mappel/PoissonNoise2D.h"
```

#### Classes

- class [PoissonGaussianNoise2DObjective< ModelBase >](#)  
*A Base type for point emitter localization models that use 2d images.*

## Functions

- `template<class Model >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value, typename Model::ImageT >::type model\_image (const Model &model, const typename Model::Stencil &s)`
- `template<class Model, class rng_t >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value, typename Model::ImageT >::type simulate\_image (const Model &model, const typename Model::Stencil &s, rng_t &rng)`  
*Simulate an image using the PSF model, by generating Poisson noise.*
- `template<class Model, class rng_t >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value, typename Model::ImageT >::type simulate\_image (const Model &model, const typename Model::ImageT &model_im, rng_t &rng)`
- `template<class Model >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value >::type model\_grad (const Model &model, const typename Model::ImageT &im, const typename Model::Stencil &s, typename Model::ParamT &grad)`
- `template<class Model >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value >::type model\_grad2 (const Model &model, const typename Model::ImageT &im, const typename Model::Stencil &s, typename Model::ParamT &grad, typename Model::ParamT &grad2)`
- `template<class Model >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value >::type model\_hessian (const Model &model, const typename Model::ImageT &im, const typename Model::Stencil &s, typename Model::ParamT &grad, typename Model::MatT &hess)`
- `template<class Model >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value, double >::type log\_likelihood (const Model &model, const typename Model::ImageT &data_im, const typename Model::Stencil &s)`
- `template<class Model >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value, double >::type relative\_log\_likelihood (const Model &model, const typename Model::ImageT &data_im, const typename Model::Stencil &s)`
- `template<class Model >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value, typename Model::MatT >::type fisher\_information (const Model &model, const typename Model::Stencil &s)`
- `template<class Model >`  
`std::enable_if< std::is_base_of< PoissonGaussianNoise2DObjective, Model >::value, std::shared_ptr< Estimator< Model > > >::type make\_estimator (const Model &model, std::string ename)`

### 10.63.1 Detailed Description

The class declaration and inline and templated functions for [PoissonGaussianNoise2DObjective](#).

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

04-2017

## 10.63.2 Function Documentation

10.63.2.1 `template<class Model > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model><←  
::value,typename Model::MatT>::type fisher_information ( const Model & model, const typename Model::Stencil & s  
)`

Definition at line 200 of file PoissonGaussianNoise2DObjective.h.

10.63.2.2 `template<class Model > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model><←  
::value,double>::type log_likelihood ( const Model & model, const typename Model::ImageT & data_im, const typename  
Model::Stencil & s )`

Definition at line 172 of file PoissonGaussianNoise2DObjective.h.

References `mappel::methods::objective::llh()`, and `mappel::poisson_log_likelihood()`.

10.63.2.3 `template<class Model > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model><←  
::value,std::shared_ptr<Estimator<Model> > >::type make_estimator ( const Model & model, std::string ename  
)`

Definition at line 217 of file PoissonGaussianNoise2DObjective.h.

References `mappel::istarts_with()`.

10.63.2.4 `template<class Model > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model>::value><←  
::type model_grad ( const Model & model, const typename Model::ImageT & im, const typename Model::Stencil & s,  
typename Model::ParamT & grad )`

Definition at line 101 of file PoissonGaussianNoise2DObjective.h.

10.63.2.5 `template<class Model > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model>::value><←  
::type model_grad2 ( const Model & model, const typename Model::ImageT & im, const typename Model::Stencil & s,  
typename Model::ParamT & grad, typename Model::ParamT & grad2 )`

Definition at line 119 of file PoissonGaussianNoise2DObjective.h.

10.63.2.6 `template<class Model > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model>::value><←  
::type model_hessian ( const Model & model, const typename Model::ImageT & im, const typename Model::Stencil & s,  
typename Model::ParamT & grad, typename Model::MatT & hess )`

Definition at line 148 of file PoissonGaussianNoise2DObjective.h.

10.63.2.7 `template<class Model > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model><←  
::value,typename Model::ImageT>::type model_image ( const Model & model, const typename Model::Stencil & s  
)`

Definition at line 59 of file PoissonGaussianNoise2DObjective.h.

10.63.2.8 `template<class Model > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model>::value,double>::type relative_log_likelihood ( const Model & model, const typename Model::ImageT & data_im, const typename Model::Stencil & s )`

Definition at line 185 of file PoissonGaussianNoise2DObjective.h.

References `mappel::relative_poisson_log_likelihood()`, and `mappel::methods::objective::rllh()`.

10.63.2.9 `template<class Model , class rng_t > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model>::value,typename Model::ImageT>::type simulate_image ( const Model & model, const typename Model::Stencil & s, rng_t & rng )`

Simulate an image using the PSF model, by generating Poisson noise.

#### Parameters

out	<i>image</i>	An image to populate.
in	<i>theta</i>	The parameter values to us
in, out	<i>rng</i>	An initialized random number generator

Definition at line 78 of file PoissonGaussianNoise2DObjective.h.

References `mappel::generate_poisson()`.

10.63.2.10 `template<class Model , class rng_t > std::enable_if<std::is_base_of<PoissonGaussianNoise2DObjective,Model>::value,typename Model::ImageT>::type simulate_image ( const Model & model, const typename Model::ImageT & model_im, rng_t & rng )`

Definition at line 89 of file PoissonGaussianNoise2DObjective.h.

References `mappel::generate_poisson()`.

## 10.64 PoissonNoise1DObjective.cpp File Reference

The class definition and template Specializations for PoissonNoise1DObjective.

```
#include "Mappel/PoissonNoise1DObjective.h"
```

#### Namespaces

- [mappel](#)

### 10.64.1 Detailed Description

The class definition and template Specializations for PoissonNoise1DObjective.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.65 PoissonNoise1DObjective.h File Reference

The class declaration and inline and templated functions for PoissonNoise1DObjective.

```
#include "Mappel/ImageFormat1DBase.h"
#include "Mappel/PoissonNoise2DObjective.h"
#include "Mappel/estimator.h"
```

#### Classes

- class [mappel::PoissonNoise1DObjective](#)

*A base class for 1D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel.*

#### Namespaces

- [mappel](#)
- [mappel::methods](#)

*Templated functions for operating on a [PointEmitterModel](#).*

- [mappel::methods::likelihood](#)
- [mappel::methods::likelihood::debug](#)

## Functions

- `template<class Model , class rng_t >`  
`ReturnIfSubclassT< ModelDataT< Model >, Model, PoissonNoise1DObjective > mappel::methods::simulate_↵`  
`_image (const Model &model, const StencilT< Model > &s, rng_t &rng)`  
*Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise1DObjective](#).*
- `template<class Model , class rng_t >`  
`ReturnIfSubclassT< ModelDataT< Model >, Model, PoissonNoise1DObjective > mappel::methods::simulate_↵`  
`_image_from_model (const Model &model, const ImageT< Model > &model_im, rng_t &rng)`  
*Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise1DObjective](#).*
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, PoissonNoise1DObjective > mappel::methods::expected_information (const`  
`Model &model, const StencilT< Model > &s)`  
*Compute the expected information (Fisher information at theta). Note: Expected information is an average quantity and is independent of the data. Enabled for [PoissonNoise1DObjective](#).*
- `template<class Model >`  
`ReturnIfSubclassT< std::unique_ptr< estimator::Estimator< Model > >, Model, PoissonNoise1DObjective >`  
`mappel::methods::make_estimator (Model &model, std::string ename)`
- `template<class Model >`  
`ReturnIfSubclassT< double, Model, PoissonNoise1DObjective > mappel::methods::likelihood::llh (const Model`  
`&model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< double, Model, PoissonNoise1DObjective > mappel::methods::likelihood::rllh (const Model`  
`&model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< ParamT< Model >, Model, PoissonNoise1DObjective > mappel::methods::likelihood::grad`  
`(const Model &model, const ModelDataT< Model > &im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, PoissonNoise1DObjective > mappel::methods::likelihood::grad2 (const Model`  
`&model, const ModelDataT< Model > &im, const StencilT< Model > &s, ParamT< Model > &grad_val,`  
`ParamT< Model > &grad2_val)`
- `template<class Model >`  
`ReturnIfSubclassT< void, Model, PoissonNoise1DObjective > mappel::methods::likelihood::hessian (const`  
`Model &model, const ModelDataT< Model > &im, const StencilT< Model > &s, ParamT< Model > &grad_val,`  
`MatT &hess_val)`
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, PoissonNoise1DObjective > mappel::methods::likelihood::debug::llh_↵`  
`components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< VecT, Model, PoissonNoise1DObjective > mappel::methods::likelihood::debug::rllh_↵`  
`components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< MatT, Model, PoissonNoise1DObjective > mappel::methods::likelihood::debug::grad_↵`  
`components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model >`  
`ReturnIfSubclassT< CubeT, Model, PoissonNoise1DObjective > mappel::methods::likelihood::debug::hessian_↵`  
`_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`

### 10.65.1 Detailed Description

The class declaration and inline and templated functions for `PoissonNoise1DObjective`.



**Author**

Mark J. Olah (mjo@cs.unm DOT edu)

**Date**

2014-2019

**10.66 PoissonNoise2DObjective.cpp File Reference**

The class definition and template Specializations for PoissonNoise2DObjective.

```
#include "Mappel/PoissonNoise2DObjective.h"
```

**Namespaces**

- [mappel](#)

**10.66.1 Detailed Description**

The class definition and template Specializations for PoissonNoise2DObjective.

**Author**

Mark J. Olah (mjo@cs.unm DOT edu)

**Date**

2014-2019

**10.67 PoissonNoise2DObjective.h File Reference**

The class declaration and inline and templated functions for PoissonNoise2DObjective.

```
#include "Mappel/ImageFormat2DBase.h"  
#include "Mappel/estimator.h"
```

**Classes**

- class [mappel::PoissonNoise2DObjective](#)

*A base class for 2D objectives with Poisson read noise. This objective function and its subclasses are for models where the only source of noise is the "shot" or "counting" or Poisson noise inherent to a discrete capture of photons given a certain mean rate of incidence on each pixel.*

## Namespaces

- [mappel](#)
- [mappel::methods](#)
  - *Templated functions for operating on a [PointEmitterModel](#).*
- [mappel::methods::likelihood](#)
- [mappel::methods::likelihood::debug](#)

## Functions

- `template<class Model, class rng_t>`  
`ReturnIfSubclassT< ImageT< Model >, Model, PoissonNoise2DObjective > mappel::methods::simulate\_image`  
`(const Model &model, const StencilT< Model > &s, rng_t &rng)`  
*Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise2DObjective](#).*
- `template<class Model, class rng_t>`  
`ReturnIfSubclassT< ImageT< Model >, Model, PoissonNoise2DObjective > mappel::methods::simulate\_image\_from\_model`  
`(const Model &model, const ImageT< Model > &model_im, rng_t &rng)`  
*Simulate an image at a given theta stencil, by generating Poisson noise Enabled for [PoissonNoise2DObjective](#).*
- `template<class Model>`  
`ReturnIfSubclassT< MatT, Model, PoissonNoise2DObjective > mappel::methods::expected\_information` (const  
`Model &model, const StencilT< Model > &s)`  
*Compute the expected information (Fisher information at theta). Note: Expected information is an average quantity and is independent of the data. Enabled for [PoissonNoise2DObjective](#).*
- `template<class Model>`  
`ReturnIfSubclassT< std::unique_ptr< estimator::Estimator< Model > >, Model, PoissonNoise2DObjective > mappel::methods::make\_estimator`  
`(Model &model, std::string ename)`
- `template<class Model>`  
`ReturnIfSubclassT< double, Model, PoissonNoise2DObjective > mappel::methods::likelihood::llh` (const Model  
`&model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model>`  
`ReturnIfSubclassT< double, Model, PoissonNoise2DObjective > mappel::methods::likelihood::rllh` (const Model  
`&model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model>`  
`ReturnIfSubclassT< ParamT< Model >, Model, PoissonNoise2DObjective > mappel::methods::likelihood::grad`  
`(const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model>`  
`ReturnIfSubclassT< void, Model, PoissonNoise2DObjective > mappel::methods::likelihood::grad2` (const Model  
`&model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model > &grad_val,`  
`ParamT< Model > &grad2_val)`
- `template<class Model>`  
`ReturnIfSubclassT< void, Model, PoissonNoise2DObjective > mappel::methods::likelihood::hessian` (const  
`Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, ParamT< Model >`  
`&grad_val, MatT &hess_val)`
- `template<class Model>`  
`ReturnIfSubclassT< VecT, Model, PoissonNoise2DObjective > mappel::methods::likelihood::debug::llh\_components`  
`(const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model>`  
`ReturnIfSubclassT< VecT, Model, PoissonNoise2DObjective > mappel::methods::likelihood::debug::rllh\_components`  
`(const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`
- `template<class Model>`  
`ReturnIfSubclassT< MatT, Model, PoissonNoise2DObjective > mappel::methods::likelihood::debug::grad\_components`  
`(const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`

- `template<class Model >`  
`ReturnIfSubclassT< CubeT, Model, PoissonNoise2DObjective > mappel::methods::likelihood::debug::hessian↔`  
`\_components (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s)`

#### 10.67.1 Detailed Description

The class declaration and inline and templated functions for PoissonNoise2DObjective.

##### Author

Mark J. Olah (mjo@cs.unm DOT edu)

##### Date

2014-2019

## 10.68 README.md File Reference

## 10.69 README.md File Reference

## 10.70 rng.cpp File Reference

Global random number generator.

```
#include "Mappel/rng.h"
```

##### Namespaces

- [mappel](#)

##### Variables

- `ParallelRngManagerT` [mappel::rng\\_manager](#)

#### 10.70.1 Detailed Description

Global random number generator.

##### Author

Mark J. Olah (mjo@cs.unm DOT edu)

##### Date

2014-2019

## 10.71 rng.h File Reference

Random number generation usign sfmt.

```
#include <random>
#include "Mappel/util.h"
#include <trng/lcg64_shift.hpp>
#include <ParallelRngManager/ParallelRngManager.h>
```

### Namespaces

- [mappel](#)

### Typedefs

- using [mappel::ParallelRngGeneratorT](#) = trng::lcg64\_shift
- using [mappel::ParallelRngManagerT](#) = parallel\_rng::ParallelRngManager< ParallelRngGeneratorT >
- using [mappel::RngSeedT](#) = parallel\_rng::SeedT
- using [mappel::UniformDistT](#) = std::uniform\_real\_distribution< double >

### Functions

- template<class RngT >  
IdxT [mappel::generate\\_poisson\\_small](#) (RngT &rng, double mu)  
*Generates a single Poisson distributed int from distribution with mean mu.*
- template<class RngT >  
IdxT [mappel::generate\\_poisson\\_large](#) (RngT &rng, double mu)
- template<class RngT >  
double [mappel::generate\\_poisson](#) (RngT &rng, double mu)

#### 10.71.1 Detailed Description

Random number generation usign sfmt.

### Author

Mark J. Olah (email [mjo@cs.unm.edu](mailto:mjo@cs.unm.edu) )

### Date

2013-2019

## 10.72 stencil.cpp File Reference

The stencils for pixel based computations.

```
#include <sstream>
#include <boost/math/special_functions/erf.hpp>
#include <boost/math/distributions/chi_squared.hpp>
#include "Mappel/util.h"
#include "Mappel/stencil.h"
#include "Mappel/display.h"
```

### Namespaces

- [mappel](#)

### Functions

- double [mappel::normal\\_quantile\\_twosided](#) (double confidence)
- double [mappel::normal\\_quantile\\_onesided](#) (double confidence)
- double [mappel::chisq\\_quantile](#) (double confidence, int dof)
- double [mappel::chisq\\_quantile](#) (double confidence)
- void [mappel::fill\\_gaussian\\_stencil](#) (int size, double stencil[], double sigma)
- double [mappel::gaussian\\_convolution](#) (int x, int y, const MatT &data, const VecT &Xstencil, const VecT &Ystencil)
- void [mappel::estimate\\_gaussian\\_2Dmax](#) (const MatT &data, const VecT &Xstencil, const VecT &Ystencil, int max\_pos[], double &min\_val)
- void [mappel::refine\\_gaussian\\_2Dmax](#) (const MatT &data, const VecT &Xstencil, const VecT &Ystencil, int max\_pos[], double &min\_val)
- double [mappel::gaussian\\_3D\\_convolution](#) (int x, int y, int z, const CubeT &data, const VecFieldT &stencils)
- void [mappel::estimate\\_gaussian\\_3Dmax](#) (const CubeT &data, const VecFieldT &stencils, int max\_pos[], double &min\_val)
- void [mappel::refine\\_gaussian\\_3Dmax](#) (const CubeT &data, const VecFieldT &stencils, int max\_pos[], double &min\_val)
- double [mappel::estimate\\_background](#) (const MatT &im, const MatT &unit\_model\_im, double min\_bg)
- double [mappel::estimate\\_intensity](#) (const MatT &im, const MatT &unit\_model\_im, double bg)
- double [mappel::estimate\\_background](#) (const CubeT &im, const CubeT &unit\_model\_im)
- double [mappel::estimate\\_intensity](#) (const CubeT &im, const CubeT &unit\_model\_im, double bg)

### 10.72.1 Detailed Description

The stencils for pixel based computations.

General utilities and helpers for Mappel.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.73 stencil.h File Reference

The stencils for pixel based computations.

```
#include <cmath>
#include <climits>
#include "Mappel/util.h"
#include "Mappel/rng.h"
```

### Namespaces

- [mappel](#)

### Functions

- double [mappel::gauss\\_norm](#) (double sigma)
- void [mappel::fill\\_d\\_stencil](#) (int size, double stencil[], double theta\_x)
- void [mappel::fill\\_G\\_stencil](#) (int size, double stencil[], const double dx[], double theta\_sigma)
- void [mappel::fill\\_X\\_stencil](#) (int size, double stencil[], const double dx[], double theta\_sigma)
- void [mappel::fill\\_DX\\_stencil](#) (int size, double stencil[], const double Gx[], double theta\_sigma)
- void [mappel::fill\\_DXS\\_stencil](#) (int size, double stencil[], const double dx[], const double Gx[], double theta\_sigma)
- void [mappel::fill\\_DXS2\\_stencil](#) (int size, double stencil[], const double dx[], const double Gx[], const double D↔XS[], double theta\_sigma)
- void [mappel::fill\\_DXSX\\_stencil](#) (int size, double stencil[], const double dx[], const double Gx[], const double DX[], double theta\_sigma)
- VecT [mappel::make\\_d\\_stencil](#) (int size, double theta\_x)
- VecT [mappel::make\\_G\\_stencil](#) (int size, const VecT &dx, double theta\_sigma)
- VecT [mappel::make\\_X\\_stencil](#) (int size, const VecT &dx, double theta\_sigma)
- VecT [mappel::make\\_DX\\_stencil](#) (int size, const VecT &Gx, double theta\_sigma)
- VecT [mappel::make\\_DXS\\_stencil](#) (int size, const VecT &dx, const VecT &Gx, double theta\_sigma)
- VecT [mappel::make\\_DXS2\\_stencil](#) (int size, const VecT &dx, const VecT &Gx, const VecT &DXS, double theta↔\_sigma)
- VecT [mappel::make\\_DXSX\\_stencil](#) (int size, const VecT &dx, const VecT &Gx, const VecT &DX, double theta↔\_sigma)
- void [mappel::fill\\_gaussian\\_stencil](#) (int size, double stencil[], double sigma)
- VecT [mappel::make\\_gaussian\\_stencil](#) (int size, double sigma)
- void [mappel::estimate\\_gaussian\\_2Dmax](#) (const MatT &data, const VecT &Xstencil, const VecT &Ystencil, int max\_pos[], double &min\_val)
- void [mappel::refine\\_gaussian\\_2Dmax](#) (const MatT &data, const VecT &Xstencil, const VecT &Ystencil, int max↔\_pos[])
- double [mappel::gaussian\\_convolution](#) (int x, int y, const MatT &data, const VecT &Xstencil, const VecT &Ystencil)
- void [mappel::estimate\\_gaussian\\_3Dmax](#) (const CubeT &data, const VecFieldT &stencils, int max\_pos[], double &min\_val)
- void [mappel::refine\\_gaussian\\_3Dmax](#) (const CubeT &data, const VecFieldT &stencils, int max\_pos[])
- double [mappel::gaussian\\_3D\\_convolution](#) (int x, int y, int z, const CubeT &data, const VecFieldT &stencils)
- double [mappel::poisson\\_log\\_likelihood](#) (double model\_val, double data\_val)
- double [mappel::relative\\_poisson\\_log\\_likelihood](#) (double model\_val, double data\_val)
- double [mappel::check\\_lower\\_bound\\_hyperparameter](#) (const char \*name, double value, double lower\_bound)

- double [mappel::check\\_positive\\_hyperparameter](#) (const char \*name, double value, double hyperprior\_epsilon=1E-6)
- double [mappel::check\\_unit\\_hyperparameter](#) (const char \*name, double value, double hyperprior\_epsilon=1E-6)
- double [mappel::log\\_prior\\_beta\\_const](#) (double beta)
- double [mappel::log\\_prior\\_beta2\\_const](#) (double beta0, double beta1)
- double [mappel::log\\_prior\\_gamma\\_const](#) (double kappa, double mean)
- double [mappel::log\\_prior\\_pareto\\_const](#) (double alpha, double min)
- double [mappel::log\\_prior\\_normal\\_const](#) (double sigma)
- double [mappel::rllh\\_beta\\_prior](#) (double beta, double v, double max=1., double min=0.)
- double [mappel::rllh\\_beta2\\_prior](#) (double beta0, double beta1, double v, double max=1., double min=0.)
- double [mappel::rllh\\_gamma\\_prior](#) (double kappa, double mean, double v)
- double [mappel::rllh\\_pareto\\_prior](#) (double alpha, double v)
- double [mappel::rllh\\_normal\\_prior](#) (double mu, double sigma)
- double [mappel::beta\\_prior\\_grad](#) (double beta, double v, double max=1., double min=0.)
- double [mappel::beta2\\_prior\\_grad](#) (double beta0, double beta1, double v, double max=1., double min=0.)
- double [mappel::gamma\\_prior\\_grad](#) (double kappa, double mean, double v)
- double [mappel::pareto\\_prior\\_grad](#) (double alpha, double v)
- double [mappel::normal\\_prior\\_grad](#) (double mu, double sigma)
- double [mappel::beta\\_prior\\_grad2](#) (double beta, double v, double max=1., double min=0.)
- double [mappel::beta2\\_prior\\_grad2](#) (double beta0, double beta1, double v, double max=1., double min=0.)
- double [mappel::gamma\\_prior\\_grad2](#) (double kappa, double v)
- double [mappel::pareto\\_prior\\_grad2](#) (double alpha, double v)
- double [mappel::normal\\_prior\\_grad](#) (double sigma)
- double [mappel::normal\\_quantile\\_twosided](#) (double confidence)
- double [mappel::normal\\_quantile\\_onesided](#) (double confidence)
- double [mappel::chisq\\_quantile](#) (double confidence, int dof)
- double [mappel::chisq\\_quantile](#) (double confidence)
- double [mappel::rllh\\_normal\\_prior](#) (double mu, double sigma, double v)
- double [mappel::normal\\_prior\\_grad](#) (double mu, double sigma, double v)
- double [mappel::normal\\_prior\\_grad2](#) (double sigma)

### 10.73.1 Detailed Description

The stencils for pixel based computations.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## 10.74 util.cpp File Reference

```
#include <sched.h>
#include <cctype>
#include <omp.h>
#include "Mappel/util.h"
```

## Namespaces

- [mappel](#)

## Functions

- void [mappel::enable\\_all\\_cpus](#) ()
- bool [mappel::istarts\\_with](#) (const char \*s, const char \*pattern)
- bool [mappel::istarts\\_with](#) (const std::string &str, const char \*pattern)
- const char \* [mappel::icontains](#) (const char \*s, const char \*pattern)
- int [mappel::maxidx](#) (const VecT &v)
- std::ostream & [mappel::operator<<](#) (std::ostream &out, const StatsT &stats)

## 10.75 util.h File Reference

Common utilities and errors.

```
#include <cstdint>
#include <cmath>
#include <memory>
#include <utility>
#include <string>
#include <map>
#include <sstream>
#include <armadillo>
#include <BacktraceException/BacktraceException.h>
```

## Classes

- struct [mappel::ParameterValueError](#)  
*Parameter value is not valid.*
- struct [mappel::ArrayShapeError](#)  
*Array is not of the right dimensionality.*
- struct [mappel::ArraySizeError](#)  
*Array is not of the right size.*
- struct [mappel::ModelBoundsError](#)  
*Access outside the model bounds is attempted.*
- struct [mappel::NumericalError](#)  
*Expected numerical condition does not hold.*
- struct [mappel::LogicalError](#)  
*Failure of code or algorithm logic.*
- struct [mappel::NotImplementedError](#)  
*Feature not yet implemented.*



## Namespaces

- [mappel](#)

## Typedefs

- using [mappel::BoolT](#) = uint16\_t
- using [mappel::BoolVecT](#) = arma::Col< uint16\_t >
- using [mappel::IdxT](#) = arma::uword
- using [mappel::IdxVecT](#) = arma::Col< IdxT >
- using [mappel::IdxMatT](#) = arma::Mat< IdxT >
- using [mappel::VecT](#) = arma::vec
- using [mappel::MatT](#) = arma::mat
- using [mappel::CubeT](#) = arma::cube
- using [mappel::VecFieldT](#) = arma::field< VecT >
- using [mappel::StatsT](#) = std::map< std::string, double >
- using [mappel::StringVecT](#) = std::vector< std::string >
- template<class ModelT, class ModelBaseT >  
using [mappel::EnableIfSubclassT](#) = typename std::enable\_if< std::is\_base\_of< ModelBaseT, ModelT >::value, void >::type
- template<class ReturnT, class ModelT, class ModelBaseT >  
using [mappel::ReturnIfSubclassT](#) = typename std::enable\_if< std::is\_base\_of< ModelBaseT, ModelT >::value, ReturnT >::type
- template<class Model >  
using [mappel::ImageCoordT](#) = typename Model::ImageCoordT
- template<class Model >  
using [mappel::ImagePixelT](#) = typename Model::ImagePixelT
- template<class Model >  
using [mappel::ParamT](#) = typename Model::ParamT
- template<class Model >  
using [mappel::ParamVecT](#) = typename Model::ParamVecT
- template<class Model >  
using [mappel::ImageT](#) = typename Model::ImageT
- template<class Model >  
using [mappel::ModelDataT](#) = typename Model::ModelDataT
- template<class Model >  
using [mappel::StencilT](#) = typename Model::Stencil
- template<class Model >  
using [mappel::ImageStackT](#) = typename Model::ImageStackT
- template<class Model >  
using [mappel::ModelDataStackT](#) = typename Model::ModelDataStackT
- template<class Model >  
using [mappel::StencilVecT](#) = typename Model::StencilVecT
- using [mappel::MappelError](#) = backtrace\_exception::BacktraceException

## Functions

- void `mappel::enable_all_cpus` ()
- bool `mappel::istarts_with` (const char \*s, const char \*pattern)
- bool `mappel::istarts_with` (const std::string &str, const char \*pattern)
- const char \* `mappel::icontains` (const char \*s, const char \*pattern)
- int `mappel::maxidx` (const VecT &v)
- template<typename T >  
int `mappel::sgn` (T val)  
*sign (signum) function: -1/0/1*
- template<typename T >  
T `mappel::square` (T x)
- double `mappel::restrict_value_range` (double val, double minval, double maxval)
- template<typename T , typename... Args>  
std::unique\_ptr< T > `mappel::make_unique` (Args &&...args)
- std::ostream & `mappel::operator<<` (std::ostream &out, const StatsT &stats)

### 10.75.1 Detailed Description

Common utilities and errors.

#### Author

Mark J. Olah (mjo@cs.unm DOT edu)

#### Date

2014-2019

## Index

- ~Estimator
  - mappel::estimator::Estimator, [119](#)
- Abort
  - omp\_exception\_catcher, [91](#)
- aposteriori\_objective
  - mappel::methods, [59](#)
- ArrayShapeError
  - mappel::ArrayShapeError, [92](#)
- ArraySizeError
  - mappel::ArraySizeError, [92](#)
- backtrack
  - mappel::estimator::IterativeMaximizer, [559](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [702](#)
  - mappel::estimator::NewtonMaximizer, [724](#)
  - mappel::estimator::QuasiNewtonMaximizer, [787](#)
  - mappel::estimator::TrustRegionMaximizer, [857](#)
- backtrack\_idxxs
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, [598](#)
- backtrack\_max\_ratio
  - mappel::estimator::IterativeMaximizer, [572](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [716](#)
  - mappel::estimator::NewtonMaximizer, [737](#)
  - mappel::estimator::QuasiNewtonMaximizer, [801](#)
  - mappel::estimator::TrustRegionMaximizer, [871](#)
- backtrack\_min\_linear\_step\_ratio
  - mappel::estimator::IterativeMaximizer, [572](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [716](#)
  - mappel::estimator::NewtonMaximizer, [737](#)
  - mappel::estimator::QuasiNewtonMaximizer, [801](#)
  - mappel::estimator::TrustRegionMaximizer, [871](#)
- backtrack\_min\_ratio
  - mappel::estimator::IterativeMaximizer, [573](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [716](#)
  - mappel::estimator::NewtonMaximizer, [737](#)
  - mappel::estimator::QuasiNewtonMaximizer, [801](#)
  - mappel::estimator::TrustRegionMaximizer, [871](#)
- beta2\_prior\_grad
  - mappel, [29](#)
- beta2\_prior\_grad2
  - mappel, [29](#)
- beta\_prior\_grad
  - mappel, [29](#)
- beta\_prior\_grad2
  - mappel, [29](#)
- bg
  - mappel::Gauss1DModel::Stencil, [838](#)
  - mappel::Gauss1DsModel::Stencil, [824](#)
  - mappel::Gauss2DModel::Stencil, [828](#)
  - mappel::Gauss2DsModel::Stencil, [832](#)
  - mappel::Gauss2DsxyModel::Stencil, [819](#)
- BoolVecT
  - mappel, [26](#)
- BoolT
  - mappel, [26](#)
- bound\_step
  - mappel::estimator::subroutine, [51](#)
- bound\_theta
  - mappel::Gauss1DModel, [135](#)
  - mappel::Gauss1DMLE, [163](#)
  - mappel::Gauss1DModel, [190](#)
  - mappel::Gauss1DsMAP, [218](#)
  - mappel::Gauss1DsMLE, [246](#)
  - mappel::Gauss1DsModel, [274](#)
  - mappel::Gauss2DModel, [303](#)
  - mappel::Gauss2DMLE, [333](#)
  - mappel::Gauss2DModel, [362](#)
  - mappel::Gauss2DsMAP, [392](#)
  - mappel::Gauss2DsMLE, [424](#)
  - mappel::Gauss2DsModel, [456](#)
  - mappel::Gauss2DsxyMAP, [486](#)
  - mappel::Gauss2DsxyModel, [513](#)
  - mappel::MAPEstimator, [580](#)
  - mappel::MCMCAdaptor1Ds, [623](#)
  - mappel::MCMCAdaptor1D, [605](#)
  - mappel::MCMCAdaptor2Ds, [659](#)
  - mappel::MCMCAdaptor2D, [641](#)
  - mappel::MLEstimator, [686](#)
  - mappel::PointEmitterModel, [749](#)
- bounded\_theta
  - mappel::Gauss1DModel, [135](#)
  - mappel::Gauss1DMLE, [163](#)
  - mappel::Gauss1DModel, [190](#)
  - mappel::Gauss1DsMAP, [218](#)
  - mappel::Gauss1DsMLE, [246](#)
  - mappel::Gauss1DsModel, [274](#)
  - mappel::Gauss2DModel, [303](#)
  - mappel::Gauss2DMLE, [333](#)
  - mappel::Gauss2DModel, [362](#)
  - mappel::Gauss2DsMAP, [392](#)
  - mappel::Gauss2DsMLE, [424](#)
  - mappel::Gauss2DsModel, [456](#)
  - mappel::Gauss2DsxyMAP, [486](#)
  - mappel::Gauss2DsxyModel, [513](#)
  - mappel::MAPEstimator, [580](#)
  - mappel::MCMCAdaptor1Ds, [623](#)
  - mappel::MCMCAdaptor1D, [605](#)
  - mappel::MCMCAdaptor2Ds, [659](#)
  - mappel::MCMCAdaptor2D, [641](#)
  - mappel::MLEstimator, [686](#)
  - mappel::PointEmitterModel, [749](#)

- bounded\_theta\_stack
  - mappel::Gauss1DMap, 135
  - mappel::Gauss1DMLE, 163
  - mappel::Gauss1DModel, 191
  - mappel::Gauss1DsMap, 218
  - mappel::Gauss1DsMLE, 246
  - mappel::Gauss1DsModel, 274
  - mappel::Gauss2DMap, 303
  - mappel::Gauss2DMLE, 333
  - mappel::Gauss2DModel, 362
  - mappel::Gauss2DsMap, 392
  - mappel::Gauss2DsMLE, 425
  - mappel::Gauss2DsModel, 456
  - mappel::Gauss2DsxyMap, 487
  - mappel::Gauss2DsxyModel, 513
  - mappel::MAPEstimator, 581
  - mappel::MCMCAdaptor1Ds, 623
  - mappel::MCMCAdaptor1D, 605
  - mappel::MCMCAdaptor2Ds, 659
  - mappel::MCMCAdaptor2D, 641
  - mappel::MLEstimator, 687
  - mappel::PointEmitterModel, 750
- bounds\_epsilon
  - mappel::Gauss1DMap, 150
  - mappel::Gauss1DMLE, 178
  - mappel::Gauss1DModel, 205
  - mappel::Gauss1DsMap, 233
  - mappel::Gauss1DsMLE, 262
  - mappel::Gauss1DsModel, 290
  - mappel::Gauss2DMap, 318
  - mappel::Gauss2DMLE, 348
  - mappel::Gauss2DModel, 378
  - mappel::Gauss2DsMap, 410
  - mappel::Gauss2DsMLE, 442
  - mappel::Gauss2DsModel, 473
  - mappel::Gauss2DsxyMap, 501
  - mappel::Gauss2DsxyModel, 525
  - mappel::MAPEstimator, 589
  - mappel::MCMCAdaptor1Ds, 633
  - mappel::MCMCAdaptor1D, 615
  - mappel::MCMCAdaptor2Ds, 669
  - mappel::MCMCAdaptor2D, 651
  - mappel::MLEstimator, 695
  - mappel::PointEmitterModel, 758
- CGaussHeuristicEstimator
  - mappel::estimator::CGaussHeuristicEstimator, 94
- CGaussMLE
  - mappel::estimator::CGaussMLE, 106
- candidate
  - mappel::mcmc::MCMCDebugData, 682
- candidate\_rllh
  - mappel::mcmc::MCMCDebugData, 682
- cgauss\_compute\_estimate
  - mappel, 29, 30
- cgauss\_compute\_estimate\_debug
  - mappel, 30
- cgauss\_heuristic\_compute\_estimate
  - mappel, 30, 31
- check\_image\_shape
  - mappel::Gauss1DMap, 136
  - mappel::Gauss1DMLE, 164
  - mappel::Gauss1DModel, 191
  - mappel::Gauss1DsMap, 218
  - mappel::Gauss1DsMLE, 247
  - mappel::Gauss1DsModel, 275
  - mappel::Gauss2DMap, 303
  - mappel::Gauss2DMLE, 333
  - mappel::Gauss2DModel, 363
  - mappel::Gauss2DsMap, 393
  - mappel::Gauss2DsMLE, 425
  - mappel::Gauss2DsModel, 457
  - mappel::Gauss2DsxyMap, 487
  - mappel::Gauss2DsxyModel, 513
  - mappel::ImageFormat1DBase, 546
  - mappel::ImageFormat2DBase, 553
  - mappel::PoissonNoise1DObjective, 766
  - mappel::PoissonNoise2DObjective, 773
- check\_lower\_bound\_hyperparameter
  - mappel, 31
- check\_param\_shape
  - mappel::Gauss1DMap, 136
  - mappel::Gauss1DMLE, 164
  - mappel::Gauss1DModel, 191
  - mappel::Gauss1DsMap, 219
  - mappel::Gauss1DsMLE, 247
  - mappel::Gauss1DsModel, 275
  - mappel::Gauss2DMap, 303, 304
  - mappel::Gauss2DMLE, 333, 334
  - mappel::Gauss2DModel, 363
  - mappel::Gauss2DsMap, 393
  - mappel::Gauss2DsMLE, 425
  - mappel::Gauss2DsModel, 457
  - mappel::Gauss2DsxyMap, 487
  - mappel::Gauss2DsxyModel, 513, 514
  - mappel::MAPEstimator, 581
  - mappel::MCMCAdaptor1Ds, 623
  - mappel::MCMCAdaptor1D, 605, 606
  - mappel::MCMCAdaptor2Ds, 660
  - mappel::MCMCAdaptor2D, 641
  - mappel::MLEstimator, 687
  - mappel::PointEmitterModel, 750
- check\_positive\_hyperparameter
  - mappel, 31
- check\_psf\_sigma
  - mappel::Gauss1DMap, 136
  - mappel::Gauss1DMLE, 164
  - mappel::Gauss1DModel, 191, 192

- mappel::Gauss1DsMAP, [219](#)
- mappel::Gauss1DsMLE, [247](#)
- mappel::Gauss1DsModel, [275](#)
- mappel::Gauss2DMAP, [304](#)
- mappel::Gauss2DMLE, [334](#)
- mappel::Gauss2DModel, [363](#)
- mappel::Gauss2DsMAP, [393](#)
- mappel::Gauss2DsMLE, [425](#), [426](#)
- mappel::Gauss2DsModel, [457](#)
- mappel::Gauss2DsxyMAP, [487](#), [488](#)
- mappel::Gauss2DsxyModel, [514](#)
- mappel::MAPEstimator, [581](#)
- mappel::MCMCAdaptor1Ds, [624](#)
- mappel::MCMCAdaptor1D, [606](#)
- mappel::MCMCAdaptor2Ds, [660](#)
- mappel::MCMCAdaptor2D, [642](#)
- mappel::MLEstimator, [687](#)
- mappel::PointEmitterModel, [750](#)
- check\_size
  - mappel::Gauss1DMAP, [137](#)
  - mappel::Gauss1DMLE, [165](#)
  - mappel::Gauss1DModel, [192](#)
  - mappel::Gauss1DsMAP, [219](#)
  - mappel::Gauss1DsMLE, [248](#)
  - mappel::Gauss1DsModel, [276](#)
  - mappel::Gauss2DMAP, [304](#)
  - mappel::Gauss2DMLE, [334](#)
  - mappel::Gauss2DModel, [364](#)
  - mappel::Gauss2DsMAP, [394](#)
  - mappel::Gauss2DsMLE, [426](#)
  - mappel::Gauss2DsModel, [458](#)
  - mappel::Gauss2DsxyMAP, [488](#)
  - mappel::Gauss2DsxyModel, [514](#)
  - mappel::ImageFormat1DBase, [546](#)
  - mappel::ImageFormat2DBase, [553](#)
  - mappel::PoissonNoise1DObjective, [767](#)
  - mappel::PoissonNoise2DObjective, [774](#)
- check\_unit\_hyperparameter
  - mappel, [31](#)
- chisq\_quantile
  - mappel, [31](#)
- cholesky
  - mappel, [31](#)
- cholesky\_convert\_full\_matrix
  - mappel, [32](#)
- cholesky\_convert\_lower\_triangular
  - mappel, [32](#)
- cholesky\_make\_negative\_definite
  - mappel, [32](#)
- cholesky\_make\_positive\_definite
  - mappel, [32](#)
- cholesky\_solve
  - mappel, [32](#)
- clamp
  - mappel, [33](#)
- clear\_stats
  - mappel::estimator::CGaussHeuristicEstimator, [94](#)
  - mappel::estimator::CGaussMLE, [107](#)
  - mappel::estimator::Estimator, [119](#)
  - mappel::estimator::HeuristicEstimator, [532](#)
  - mappel::estimator::IterativeMaximizer, [559](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [702](#)
  - mappel::estimator::NewtonMaximizer, [724](#)
  - mappel::estimator::QuasiNewtonMaximizer, [787](#)
  - mappel::estimator::SimulatedAnnealingMaximizer, [807](#)
  - mappel::estimator::ThreadedEstimator, [842](#)
  - mappel::estimator::TrustRegionMaximizer, [857](#)
- ClockT
  - estimator.h, [882](#)
- compute\_D\_scale
  - mappel::estimator::subroutine, [52](#)
- compute\_bound\_scaling\_vec
  - mappel::estimator::subroutine, [52](#)
- compute\_cauchy\_point
  - mappel::estimator::subroutine, [52](#)
- compute\_derivatives
  - mappel::Gauss1DModel::Stencil, [838](#)
  - mappel::Gauss1DsModel::Stencil, [824](#)
  - mappel::Gauss2DModel::Stencil, [828](#)
  - mappel::Gauss2DsModel::Stencil, [832](#)
  - mappel::Gauss2DsxyModel::Stencil, [819](#)
- compute\_estimate
  - mappel::estimator::Estimator, [119](#)
  - mappel::estimator::IterativeMaximizer, [560](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [702](#)
  - mappel::estimator::NewtonMaximizer, [724](#)
  - mappel::estimator::QuasiNewtonMaximizer, [787](#)
  - mappel::estimator::ThreadedEstimator, [842](#)
  - mappel::estimator::TrustRegionMaximizer, [857](#)
- compute\_estimate\_debug
  - mappel::estimator::CGaussHeuristicEstimator, [94](#)
  - mappel::estimator::Estimator, [120](#)
  - mappel::estimator::HeuristicEstimator, [532](#)
  - mappel::estimator::IterativeMaximizer, [560](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [703](#)
  - mappel::estimator::NewtonMaximizer, [724](#)
  - mappel::estimator::QuasiNewtonMaximizer, [788](#)
  - mappel::estimator::ThreadedEstimator, [843](#)
  - mappel::estimator::TrustRegionMaximizer, [858](#)
- compute\_initial\_trust\_radius
  - mappel::estimator::subroutine, [52](#)
- compute\_max\_sigma\_ratio
  - mappel::Gauss2DsMAP, [394](#)
  - mappel::Gauss2DsMLE, [426](#)
  - mappel::Gauss2DsModel, [458](#)
  - mappel::Gauss2DsxyModel, [514](#)
- compute\_posterior\_credible

- mappel::mcmc, [54](#)
- compute\_profile\_bound
  - mappel::estimator::CGaussHeuristicEstimator, [95](#)
  - mappel::estimator::CGaussMLE, [107](#)
  - mappel::estimator::Estimator, [120](#)
  - mappel::estimator::HeuristicEstimator, [533](#)
  - mappel::estimator::IterativeMaximizer, [560](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [703](#)
  - mappel::estimator::NewtonMaximizer, [725](#)
  - mappel::estimator::QuasiNewtonMaximizer, [788](#)
  - mappel::estimator::SimulatedAnnealingMaximizer, [807](#)
  - mappel::estimator::ThreadedEstimator, [843](#)
  - mappel::estimator::TrustRegionMaximizer, [858](#)
- compute\_profile\_bound\_debug
  - mappel::estimator::CGaussHeuristicEstimator, [95](#)
  - mappel::estimator::CGaussMLE, [107](#)
  - mappel::estimator::Estimator, [120](#)
  - mappel::estimator::HeuristicEstimator, [533](#)
  - mappel::estimator::IterativeMaximizer, [561](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [703](#)
  - mappel::estimator::NewtonMaximizer, [725](#)
  - mappel::estimator::QuasiNewtonMaximizer, [788](#)
  - mappel::estimator::SimulatedAnnealingMaximizer, [808](#)
  - mappel::estimator::ThreadedEstimator, [843](#)
  - mappel::estimator::TrustRegionMaximizer, [858](#)
- compute\_profile\_estimate
  - mappel::estimator::CGaussHeuristicEstimator, [95](#)
  - mappel::estimator::CGaussMLE, [107](#)
  - mappel::estimator::Estimator, [120](#)
  - mappel::estimator::HeuristicEstimator, [533](#)
  - mappel::estimator::IterativeMaximizer, [561](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [704](#)
  - mappel::estimator::NewtonMaximizer, [725](#)
  - mappel::estimator::QuasiNewtonMaximizer, [789](#)
  - mappel::estimator::ThreadedEstimator, [843](#)
  - mappel::estimator::TrustRegionMaximizer, [859](#)
- compute\_quadratic\_model\_value
  - mappel::estimator::subroutine, [53](#)
- compute\_scaled\_problem
  - mappel::estimator::subroutine, [53](#)
- confidence
  - mappel::estimator::ProfileBoundsData, [778](#)
  - mappel::estimator::ProfileBoundsDataStack, [782](#)
  - mappel::mcmc::MCMCData, [678](#)
  - mappel::mcmc::MCMCDataStack, [680](#)
- Continue
  - omp\_exception\_catcher, [91](#)
- convergence\_min\_function\_change\_ratio
  - mappel::estimator::IterativeMaximizer, [573](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [716](#)
  - mappel::estimator::NewtonMaximizer, [737](#)
  - mappel::estimator::QuasiNewtonMaximizer, [801](#)
- mappel::estimator::TrustRegionMaximizer, [871](#)
- convergence\_min\_step\_size\_ratio
  - mappel::estimator::IterativeMaximizer, [573](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [716](#)
  - mappel::estimator::NewtonMaximizer, [737](#)
  - mappel::estimator::QuasiNewtonMaximizer, [801](#)
  - mappel::estimator::TrustRegionMaximizer, [871](#)
- convergence\_min\_trust\_radius
  - mappel::estimator::TrustRegionMaximizer, [872](#)
- convergence\_test\_grad\_ratio
  - mappel::estimator::IterativeMaximizer, [561](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [704](#)
  - mappel::estimator::NewtonMaximizer, [726](#)
  - mappel::estimator::QuasiNewtonMaximizer, [789](#)
  - mappel::estimator::TrustRegionMaximizer, [859](#)
- convergence\_test\_step\_size
  - mappel::estimator::IterativeMaximizer, [562](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [704](#)
  - mappel::estimator::NewtonMaximizer, [726](#)
  - mappel::estimator::QuasiNewtonMaximizer, [789](#)
  - mappel::estimator::TrustRegionMaximizer, [859](#)
- CoordIdxT
  - PoissonGaussianNoise2DObjective, [761](#)
- CoordStackT
  - PoissonGaussianNoise2DObjective, [761](#)
- CoordT
  - PoissonGaussianNoise2DObjective, [761](#)
- copy\_Lsym\_mat
  - mappel, [33](#)
- copy\_Usym\_mat
  - mappel, [33](#)
- copy\_Usym\_mat\_stack
  - mappel, [33](#)
- cr\_lower\_bound
  - mappel::methods, [59](#)
- cr\_lower\_bound\_stack
  - mappel::methods::openmp, [85](#)
- credible\_lb
  - mappel::mcmc::MCMCData, [678](#)
  - mappel::mcmc::MCMCDataStack, [680](#)
- credible\_ub
  - mappel::mcmc::MCMCData, [678](#)
  - mappel::mcmc::MCMCDataStack, [680](#)
- CubeT
  - mappel, [27](#)
- current\_stencil
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, [598](#)
- DXS2
  - mappel::Gauss1DsModel::Stencil, [825](#)
  - mappel::Gauss2DsModel::Stencil, [834](#)
  - mappel::Gauss2DsxyModel::Stencil, [820](#)
- DXSX

- mappel::Gauss1DsModel::Stencil, [825](#)
- mappel::Gauss2DsModel::Stencil, [835](#)
- mappel::Gauss2DsxyModel::Stencil, [820](#)
- DXS
  - mappel::Gauss1DModel::Stencil, [839](#)
  - mappel::Gauss1DsModel::Stencil, [825](#)
  - mappel::Gauss2DModel::Stencil, [829](#)
  - mappel::Gauss2DsModel::Stencil, [834](#)
  - mappel::Gauss2DsxyModel::Stencil, [820](#)
- DYS2
  - mappel::Gauss2DsModel::Stencil, [835](#)
  - mappel::Gauss2DsxyModel::Stencil, [821](#)
- DYSX
  - mappel::Gauss2DsxyModel::Stencil, [821](#)
- DYSY
  - mappel::Gauss2DsModel::Stencil, [835](#)
  - mappel::Gauss2DsxyModel::Stencil, [821](#)
- DYS
  - mappel::Gauss2DModel::Stencil, [830](#)
  - mappel::Gauss2DsModel::Stencil, [835](#)
  - mappel::Gauss2DsxyModel::Stencil, [821](#)
- debug\_internal\_sum\_model\_x
  - mappel::Gauss2DMAP, [304](#)
  - mappel::Gauss2DMLE, [334](#)
  - mappel::Gauss2DModel, [364](#)
  - mappel::Gauss2DsMAP, [394](#)
  - mappel::Gauss2DsMLE, [426](#)
  - mappel::Gauss2DsModel, [458](#)
- debug\_internal\_sum\_model\_y
  - mappel::Gauss2DMAP, [304](#)
  - mappel::Gauss2DMLE, [334](#)
  - mappel::Gauss2DModel, [364](#)
  - mappel::Gauss2DsMAP, [394](#)
  - mappel::Gauss2DsMLE, [426](#)
  - mappel::Gauss2DsModel, [458](#)
- Default\_T\_Init
  - mappel::estimator::SimulatedAnnealingMaximizer, [815](#)
- default\_alpha\_sigma
  - mappel::Gauss1DMAP, [150](#)
  - mappel::Gauss1DMLE, [178](#)
  - mappel::Gauss1DModel, [205](#)
  - mappel::Gauss1DsMAP, [233](#)
  - mappel::Gauss1DsMLE, [262](#)
  - mappel::Gauss1DsModel, [290](#)
  - mappel::Gauss2DMAP, [318](#)
  - mappel::Gauss2DMLE, [348](#)
  - mappel::Gauss2DModel, [378](#)
  - mappel::Gauss2DsMAP, [410](#)
  - mappel::Gauss2DsMLE, [442](#)
  - mappel::Gauss2DsModel, [473](#)
  - mappel::Gauss2DsxyMAP, [501](#)
  - mappel::Gauss2DsxyModel, [525](#)
  - mappel::MAPEstimator, [589](#)
- mappel::MCMCAdaptor1Ds, [633](#)
- mappel::MCMCAdaptor1D, [615](#)
- mappel::MCMCAdaptor2Ds, [669](#)
- mappel::MCMCAdaptor2D, [651](#)
- mappel::MLEstimator, [695](#)
- mappel::PointEmitterModel, [758](#)
- default\_beta\_pos
  - mappel::Gauss1DMAP, [150](#)
  - mappel::Gauss1DMLE, [178](#)
  - mappel::Gauss1DModel, [205](#)
  - mappel::Gauss1DsMAP, [234](#)
  - mappel::Gauss1DsMLE, [262](#)
  - mappel::Gauss1DsModel, [290](#)
  - mappel::Gauss2DMAP, [319](#)
  - mappel::Gauss2DMLE, [349](#)
  - mappel::Gauss2DModel, [378](#)
  - mappel::Gauss2DsMAP, [410](#)
  - mappel::Gauss2DsMLE, [442](#)
  - mappel::Gauss2DsModel, [473](#)
  - mappel::Gauss2DsxyMAP, [501](#)
  - mappel::Gauss2DsxyModel, [525](#)
  - mappel::MAPEstimator, [589](#)
  - mappel::MCMCAdaptor1Ds, [633](#)
  - mappel::MCMCAdaptor1D, [615](#)
  - mappel::MCMCAdaptor2Ds, [670](#)
  - mappel::MCMCAdaptor2D, [651](#)
  - mappel::MLEstimator, [695](#)
  - mappel::PointEmitterModel, [758](#)
- default\_intensity\_kappa
  - mappel::Gauss1DMAP, [151](#)
  - mappel::Gauss1DMLE, [179](#)
  - mappel::Gauss1DModel, [205](#)
  - mappel::Gauss1DsMAP, [234](#)
  - mappel::Gauss1DsMLE, [262](#)
  - mappel::Gauss1DsModel, [290](#)
  - mappel::Gauss2DMAP, [319](#)
  - mappel::Gauss2DMLE, [349](#)
  - mappel::Gauss2DModel, [378](#)
  - mappel::Gauss2DsMAP, [410](#)
  - mappel::Gauss2DsMLE, [442](#)
  - mappel::Gauss2DsModel, [473](#)
  - mappel::Gauss2DsxyMAP, [501](#)
  - mappel::Gauss2DsxyModel, [526](#)
  - mappel::MAPEstimator, [589](#)
  - mappel::MCMCAdaptor1Ds, [633](#)
  - mappel::MCMCAdaptor1D, [615](#)
  - mappel::MCMCAdaptor2Ds, [670](#)
  - mappel::MCMCAdaptor2D, [651](#)
  - mappel::MLEstimator, [695](#)
  - mappel::PointEmitterModel, [758](#)
- default\_max\_l
  - mappel::Gauss1DMAP, [151](#)
  - mappel::Gauss1DMLE, [179](#)
  - mappel::Gauss1DModel, [205](#)



- mappel::Gauss1DsMAP, [234](#)
- mappel::Gauss1DsMLE, [262](#)
- mappel::Gauss1DsModel, [290](#)
- mappel::Gauss2DMAP, [319](#)
- mappel::Gauss2DMLE, [349](#)
- mappel::Gauss2DModel, [378](#)
- mappel::Gauss2DsMAP, [410](#)
- mappel::Gauss2DsMLE, [442](#)
- mappel::Gauss2DsModel, [474](#)
- mappel::Gauss2DsxyMAP, [501](#)
- mappel::Gauss2DsxyModel, [526](#)
- mappel::MAPEstimator, [589](#)
- mappel::MCMCAdaptor1Ds, [633](#)
- mappel::MCMCAdaptor1D, [615](#)
- mappel::MCMCAdaptor2Ds, [670](#)
- mappel::MCMCAdaptor2D, [651](#)
- mappel::MLEstimator, [695](#)
- mappel::PointEmitterModel, [758](#)
- default\_mean\_l
  - mappel::Gauss1DMAP, [151](#)
  - mappel::Gauss1DMLE, [179](#)
  - mappel::Gauss1DModel, [206](#)
  - mappel::Gauss1DsMAP, [234](#)
  - mappel::Gauss1DsMLE, [262](#)
  - mappel::Gauss1DsModel, [290](#)
  - mappel::Gauss2DMAP, [319](#)
  - mappel::Gauss2DMLE, [349](#)
  - mappel::Gauss2DModel, [378](#)
  - mappel::Gauss2DsMAP, [410](#)
  - mappel::Gauss2DsMLE, [442](#)
  - mappel::Gauss2DsModel, [474](#)
  - mappel::Gauss2DsxyMAP, [502](#)
  - mappel::Gauss2DsxyModel, [526](#)
  - mappel::MAPEstimator, [590](#)
  - mappel::MCMCAdaptor1Ds, [633](#)
  - mappel::MCMCAdaptor1D, [616](#)
  - mappel::MCMCAdaptor2Ds, [670](#)
  - mappel::MCMCAdaptor2D, [651](#)
  - mappel::MLEstimator, [696](#)
  - mappel::PointEmitterModel, [758](#)
- default\_pixel\_mean\_bg
  - mappel::Gauss1DMAP, [151](#)
  - mappel::Gauss1DMLE, [179](#)
  - mappel::Gauss1DModel, [206](#)
  - mappel::Gauss1DsMAP, [234](#)
  - mappel::Gauss1DsMLE, [262](#)
  - mappel::Gauss1DsModel, [290](#)
  - mappel::Gauss2DMAP, [319](#)
  - mappel::Gauss2DMLE, [349](#)
  - mappel::Gauss2DModel, [379](#)
  - mappel::Gauss2DsMAP, [410](#)
  - mappel::Gauss2DsMLE, [443](#)
  - mappel::Gauss2DsModel, [474](#)
  - mappel::Gauss2DsxyMAP, [502](#)
- mappel::Gauss2DsxyModel, [526](#)
- mappel::MAPEstimator, [590](#)
- mappel::MCMCAdaptor1Ds, [633](#)
- mappel::MCMCAdaptor1D, [616](#)
- mappel::MCMCAdaptor2Ds, [670](#)
- mappel::MCMCAdaptor2D, [651](#)
- mappel::MLEstimator, [696](#)
- mappel::PointEmitterModel, [758](#)
- default\_sigma\_pos
  - mappel::Gauss1DMAP, [151](#)
  - mappel::Gauss1DMLE, [179](#)
  - mappel::Gauss1DModel, [206](#)
  - mappel::Gauss1DsMAP, [234](#)
  - mappel::Gauss1DsMLE, [263](#)
  - mappel::Gauss1DsModel, [291](#)
  - mappel::Gauss2DMAP, [319](#)
  - mappel::Gauss2DMLE, [349](#)
  - mappel::Gauss2DModel, [379](#)
  - mappel::Gauss2DsMAP, [411](#)
  - mappel::Gauss2DsMLE, [443](#)
  - mappel::Gauss2DsModel, [474](#)
  - mappel::Gauss2DsxyMAP, [502](#)
  - mappel::Gauss2DsxyModel, [526](#)
  - mappel::MAPEstimator, [590](#)
  - mappel::MCMCAdaptor1Ds, [634](#)
  - mappel::MCMCAdaptor1D, [616](#)
  - mappel::MCMCAdaptor2Ds, [670](#)
  - mappel::MCMCAdaptor2D, [652](#)
  - mappel::MLEstimator, [696](#)
  - mappel::PointEmitterModel, [759](#)
- DefaultCoolingRate
  - mappel::estimator::SimulatedAnnealingMaximizer, [815](#)
- DefaultIterations
  - mappel::estimator::CGaussMLE, [115](#)
  - mappel::estimator::IterativeMaximizer, [573](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [717](#)
  - mappel::estimator::NewtonMaximizer, [738](#)
  - mappel::estimator::QuasiNewtonMaximizer, [802](#)
  - mappel::estimator::TrustRegionMaximizer, [872](#)
- DefaultMaxSeqLength
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, [598](#)
- DefaultNumIterations
  - mappel::estimator::SimulatedAnnealingMaximizer, [815](#)
- DefaultPriorType
  - mappel::Gauss1DMAP, [151](#)
  - mappel::Gauss1DMLE, [179](#)
  - mappel::Gauss1DModel, [206](#)
  - mappel::Gauss1DsMAP, [235](#)
  - mappel::Gauss1DsMLE, [263](#)
  - mappel::Gauss1DsModel, [291](#)
  - mappel::Gauss2DMAP, [320](#)



- mappel::Gauss2DMLE, 350
- mappel::Gauss2DModel, 379
- mappel::Gauss2DsMAP, 411
- mappel::Gauss2DsMLE, 443
- mappel::Gauss2DsModel, 474
- mappel::Gauss2DsxyMAP, 502
- DefaultSeperableInitEstimator
  - mappel::Gauss1DModel, 152
  - mappel::Gauss1DMLE, 180
  - mappel::Gauss1DModel, 206
  - mappel::Gauss1DsMAP, 235
  - mappel::Gauss1DsMLE, 263
  - mappel::Gauss1DsModel, 291
  - mappel::Gauss2DModel, 320
  - mappel::Gauss2DMLE, 350
  - mappel::Gauss2DModel, 379
  - mappel::Gauss2DsMAP, 411
  - mappel::Gauss2DsMLE, 443
  - mappel::Gauss2DsModel, 474
  - mappel::Gauss2DsxyMAP, 502
  - mappel::Gauss2DsxyModel, 526
  - mappel::MAPEstimator, 590
  - mappel::MCMCAdaptor1Ds, 634
  - mappel::MCMCAdaptor1D, 616
  - mappel::MCMCAdaptor2Ds, 671
  - mappel::MCMCAdaptor2D, 652
  - mappel::MLEstimator, 696
  - mappel::PointEmitterModel, 759
- derivatives\_computed
  - mappel::Gauss1DModel::Stencil, 839
  - mappel::Gauss1DsModel::Stencil, 825
  - mappel::Gauss2DModel::Stencil, 829
  - mappel::Gauss2DsModel::Stencil, 834
  - mappel::Gauss2DsxyModel::Stencil, 820
- display.cpp, 876
- display.h, 877
- DoNotTry
  - omp\_exception\_catcher, 91
- DX
  - mappel::Gauss1DModel::Stencil, 839
  - mappel::Gauss1DsModel::Stencil, 825
  - mappel::Gauss2DModel::Stencil, 829
  - mappel::Gauss2DsModel::Stencil, 834
  - mappel::Gauss2DsxyModel::Stencil, 820
- dx
  - mappel::Gauss1DModel::Stencil, 839
  - mappel::Gauss1DsModel::Stencil, 825
  - mappel::Gauss2DModel::Stencil, 829
  - mappel::Gauss2DsModel::Stencil, 834
  - mappel::Gauss2DsxyModel::Stencil, 820
- DY
  - mappel::Gauss2DModel::Stencil, 830
  - mappel::Gauss2DsModel::Stencil, 835
  - mappel::Gauss2DsxyModel::Stencil, 821
- dy
  - mappel::Gauss2DModel::Stencil, 829
  - mappel::Gauss2DsModel::Stencil, 835
  - mappel::Gauss2DsxyModel::Stencil, 821
- enable\_all\_cpus
  - mappel, 33
- EnableIfSubclassT
  - mappel, 27
- Error
  - mappel::estimator, 50
- error\_bounds\_expected
  - mappel::methods, 59
- error\_bounds\_expected\_stack
  - mappel::methods::openmp, 86
- error\_bounds\_observed
  - mappel::methods, 59
- error\_bounds\_observed\_stack
  - mappel::methods::openmp, 86
- error\_bounds\_posterior\_credible
  - mappel::methods, 60
- error\_bounds\_profile\_likelihood
  - mappel::methods, 60
- error\_bounds\_profile\_likelihood\_debug
  - mappel::methods::debug, 67
- error\_bounds\_profile\_likelihood\_parallel
  - mappel::methods::openmp, 86
- error\_bounds\_profile\_likelihood\_stack
  - mappel::methods::openmp, 86, 87
- estimate\_background
  - mappel, 33, 34
- estimate\_gaussian\_2Dmax
  - mappel, 34
- estimate\_gaussian\_3Dmax
  - mappel, 34
- estimate\_intensity
  - mappel, 34
- estimate\_max
  - mappel::estimator::CGaussHeuristicEstimator, 95, 96
  - mappel::estimator::CGaussMLE, 107, 108
  - mappel::estimator::Estimator, 121, 122
  - mappel::estimator::HeuristicEstimator, 533, 534
  - mappel::estimator::IterativeMaximizer, 562, 563
  - mappel::estimator::NewtonDiagonalMaximizer, 704, 705
  - mappel::estimator::NewtonMaximizer, 726, 727
  - mappel::estimator::QuasiNewtonMaximizer, 789, 790
  - mappel::estimator::SimulatedAnnealingMaximizer, 808, 809
  - mappel::estimator::ThreadedEstimator, 844, 845
  - mappel::estimator::TrustRegionMaximizer, 859, 860
  - mappel::methods, 60, 61
- estimate\_max\_debug

- mappel::estimator::CGaussHeuristicEstimator, 97
- mappel::estimator::CGaussMLE, 109
- mappel::estimator::Estimator, 122, 123
- mappel::estimator::HeuristicEstimator, 535
- mappel::estimator::IterativeMaximizer, 563, 564
- mappel::estimator::NewtonDiagonalMaximizer, 706
- mappel::estimator::NewtonMaximizer, 728
- mappel::estimator::QuasiNewtonMaximizer, 791
- mappel::estimator::SimulatedAnnealingMaximizer, 809, 810
- mappel::estimator::ThreadedEstimator, 845, 846
- mappel::estimator::TrustRegionMaximizer, 861
- mappel::methods::debug, 67
- estimate\_max\_stack
  - mappel::estimator::CGaussHeuristicEstimator, 98
  - mappel::estimator::CGaussMLE, 110
  - mappel::estimator::Estimator, 123, 124
  - mappel::estimator::HeuristicEstimator, 536
  - mappel::estimator::IterativeMaximizer, 564, 565
  - mappel::estimator::NewtonDiagonalMaximizer, 707
  - mappel::estimator::NewtonMaximizer, 729
  - mappel::estimator::QuasiNewtonMaximizer, 792
  - mappel::estimator::SimulatedAnnealingMaximizer, 810, 811
  - mappel::estimator::ThreadedEstimator, 846, 847
  - mappel::estimator::TrustRegionMaximizer, 862
  - mappel::methods::openmp, 87
- estimate\_posterior
  - mappel::methods, 61
- estimate\_posterior\_debug
  - mappel::methods::debug, 68
- estimate\_posterior\_stack
  - mappel::methods::openmp, 87, 88
- estimate\_profile\_bounds
  - mappel::estimator::CGaussHeuristicEstimator, 99
  - mappel::estimator::CGaussMLE, 111
  - mappel::estimator::Estimator, 124
  - mappel::estimator::HeuristicEstimator, 537
  - mappel::estimator::IterativeMaximizer, 565
  - mappel::estimator::NewtonDiagonalMaximizer, 708
  - mappel::estimator::NewtonMaximizer, 730
  - mappel::estimator::QuasiNewtonMaximizer, 793
  - mappel::estimator::SimulatedAnnealingMaximizer, 811
  - mappel::estimator::ThreadedEstimator, 847
  - mappel::estimator::TrustRegionMaximizer, 863
- estimate\_profile\_bounds\_debug
  - mappel::estimator::CGaussHeuristicEstimator, 99
  - mappel::estimator::CGaussMLE, 111
  - mappel::estimator::Estimator, 124
  - mappel::estimator::HeuristicEstimator, 537
  - mappel::estimator::IterativeMaximizer, 566
  - mappel::estimator::NewtonDiagonalMaximizer, 708
  - mappel::estimator::NewtonMaximizer, 730
- mappel::estimator::QuasiNewtonMaximizer, 793
- mappel::estimator::SimulatedAnnealingMaximizer, 812
- mappel::estimator::ThreadedEstimator, 847
- mappel::estimator::TrustRegionMaximizer, 863
- estimate\_profile\_bounds\_parallel
  - mappel::estimator::CGaussHeuristicEstimator, 100
  - mappel::estimator::CGaussMLE, 112
  - mappel::estimator::Estimator, 125
  - mappel::estimator::HeuristicEstimator, 538
  - mappel::estimator::IterativeMaximizer, 566
  - mappel::estimator::NewtonDiagonalMaximizer, 709
  - mappel::estimator::NewtonMaximizer, 730
  - mappel::estimator::QuasiNewtonMaximizer, 794
  - mappel::estimator::SimulatedAnnealingMaximizer, 812
  - mappel::estimator::ThreadedEstimator, 848
  - mappel::estimator::TrustRegionMaximizer, 864
- estimate\_profile\_bounds\_stack
  - mappel::estimator::CGaussHeuristicEstimator, 100
  - mappel::estimator::CGaussMLE, 112
  - mappel::estimator::Estimator, 125
  - mappel::estimator::HeuristicEstimator, 538
  - mappel::estimator::IterativeMaximizer, 566
  - mappel::estimator::NewtonDiagonalMaximizer, 709
  - mappel::estimator::NewtonMaximizer, 730
  - mappel::estimator::QuasiNewtonMaximizer, 794
  - mappel::estimator::SimulatedAnnealingMaximizer, 812
  - mappel::estimator::ThreadedEstimator, 848
  - mappel::estimator::TrustRegionMaximizer, 864
- estimate\_profile\_likelihood
  - mappel::methods, 61, 62
- estimate\_profile\_likelihood\_stack
  - mappel::methods::openmp, 88
- estimate\_profile\_max
  - mappel::estimator::CGaussHeuristicEstimator, 101
  - mappel::estimator::CGaussMLE, 113
  - mappel::estimator::Estimator, 125
  - mappel::estimator::HeuristicEstimator, 539
  - mappel::estimator::IterativeMaximizer, 567
  - mappel::estimator::NewtonDiagonalMaximizer, 710
  - mappel::estimator::NewtonMaximizer, 731
  - mappel::estimator::QuasiNewtonMaximizer, 795
  - mappel::estimator::SimulatedAnnealingMaximizer, 813
  - mappel::estimator::ThreadedEstimator, 849
  - mappel::estimator::TrustRegionMaximizer, 865
- estimate\_sample\_posterior
  - mappel::mcmc, 54
- estimated\_idx
  - mappel::estimator::ProfileBoundsData, 778
  - mappel::estimator::ProfileBoundsDataStack, 782
- Estimator

- mappel::estimator::Estimator, 119
- estimator.cpp, 878
- estimator.h, 879
  - ClockT, 882
- estimator\_helpers.h, 882
- estimator\_impl.h, 883
- estimator\_names
  - mappel::Gauss1DMap, 152
  - mappel::Gauss1DMLE, 180
  - mappel::Gauss1DsMAP, 235
  - mappel::Gauss1DsMLE, 263
  - mappel::Gauss2DMap, 320
  - mappel::Gauss2DMLE, 350
  - mappel::Gauss2DsMAP, 411
  - mappel::Gauss2DsMLE, 443
  - mappel::Gauss2DsxyMAP, 502
  - mappel::PoissonNoise1DObjective, 769
  - mappel::PoissonNoise2DObjective, 776
  - PoissonGaussianNoise2DObjective, 762
- estimator\_statics.cpp, 884
- eta\_bg
  - mappel::Gauss1DMap, 152
  - mappel::Gauss1DMLE, 180
  - mappel::Gauss1DModel, 206
  - mappel::Gauss1DsMAP, 235
  - mappel::Gauss1DsMLE, 263
  - mappel::Gauss1DsModel, 291
  - mappel::Gauss2DMap, 320
  - mappel::Gauss2DMLE, 350
  - mappel::Gauss2DModel, 379
  - mappel::Gauss2DsMAP, 411
  - mappel::Gauss2DsMLE, 443
  - mappel::Gauss2DsModel, 475
  - mappel::Gauss2DsxyMAP, 503
  - mappel::MCMCAdaptor1Ds, 634
  - mappel::MCMCAdaptor1D, 616
  - mappel::MCMCAdaptor2Ds, 671
  - mappel::MCMCAdaptor2D, 652
- eta\_l
  - mappel::Gauss1DMap, 152
  - mappel::Gauss1DMLE, 180
  - mappel::Gauss1DModel, 207
  - mappel::Gauss1DsMAP, 235
  - mappel::Gauss1DsMLE, 263
  - mappel::Gauss1DsModel, 291
  - mappel::Gauss2DMap, 320
  - mappel::Gauss2DMLE, 350
  - mappel::Gauss2DModel, 379
  - mappel::Gauss2DsMAP, 411
  - mappel::Gauss2DsMLE, 444
  - mappel::Gauss2DsModel, 475
  - mappel::Gauss2DsxyMAP, 503
  - mappel::MCMCAdaptor1Ds, 634
  - mappel::MCMCAdaptor1D, 616
- mappel::MCMCAdaptor2Ds, 671
- mappel::MCMCAdaptor2D, 652
- eta\_sigma
  - mappel::Gauss1DsMAP, 235
  - mappel::Gauss1DsMLE, 264
  - mappel::Gauss1DsModel, 291
  - mappel::Gauss2DsMAP, 412
  - mappel::Gauss2DsMLE, 444
  - mappel::Gauss2DsModel, 475
  - mappel::MCMCAdaptor1Ds, 634
  - mappel::MCMCAdaptor2Ds, 671
- eta\_x
  - mappel::Gauss1DMap, 152
  - mappel::Gauss1DMLE, 180
  - mappel::Gauss1DModel, 207
  - mappel::Gauss1DsMAP, 236
  - mappel::Gauss1DsMLE, 264
  - mappel::Gauss1DsModel, 292
  - mappel::Gauss2DMap, 320
  - mappel::Gauss2DMLE, 350
  - mappel::Gauss2DModel, 380
  - mappel::Gauss2DsMAP, 412
  - mappel::Gauss2DsMLE, 444
  - mappel::Gauss2DsModel, 475
  - mappel::Gauss2DsxyMAP, 503
  - mappel::MCMCAdaptor1Ds, 635
  - mappel::MCMCAdaptor1D, 617
  - mappel::MCMCAdaptor2Ds, 671
  - mappel::MCMCAdaptor2D, 652
- eta\_y
  - mappel::Gauss2DMap, 321
  - mappel::Gauss2DMLE, 351
  - mappel::Gauss2DModel, 380
  - mappel::Gauss2DsMAP, 412
  - mappel::Gauss2DsMLE, 444
  - mappel::Gauss2DsModel, 475
  - mappel::MCMCAdaptor2Ds, 672
  - mappel::MCMCAdaptor2D, 653
- exit\_counts
  - mappel::estimator::CGaussHeuristicEstimator, 103
  - mappel::estimator::CGaussMLE, 115
  - mappel::estimator::Estimator, 127
  - mappel::estimator::HeuristicEstimator, 541
  - mappel::estimator::IterativeMaximizer, 573
  - mappel::estimator::NewtonDiagonalMaximizer, 717
  - mappel::estimator::NewtonMaximizer, 738
  - mappel::estimator::QuasiNewtonMaximizer, 802
  - mappel::estimator::SimulatedAnnealingMaximizer, 816
  - mappel::estimator::ThreadedEstimator, 851
  - mappel::estimator::TrustRegionMaximizer, 872
- ExitCode
  - mappel::estimator, 50
- expand\_max\_seq\_len

- mappel::estimator::IterativeMaximizer::Maximizer↔
  - Data, 594
- expected\_information
  - mappel::methods, 62
- expected\_information\_stack
  - mappel::methods::openmp, 88
- fill\_DX\_stencil
  - mappel, 34
- fill\_DXS2\_stencil
  - mappel, 34
- fill\_DXS\_stencil
  - mappel, 34
- fill\_DXSX\_stencil
  - mappel, 35
- fill\_G\_stencil
  - mappel, 35
- fill\_X\_stencil
  - mappel, 35
- fill\_d\_stencil
  - mappel, 34
- fill\_gaussian\_stencil
  - mappel, 35
- fisher\_information
  - PoissonGaussianNoise2DObjective.h, 931
- fixed\_idx
  - mappel::estimator::IterativeMaximizer::Maximizer↔
    - Data, 598
- free\_idx
  - mappel::estimator::IterativeMaximizer::Maximizer↔
    - Data, 599
- FunctionValue
  - mappel::estimator, 50
- gamma\_prior\_grad
  - mappel, 35
- gamma\_prior\_grad2
  - mappel, 35
- Gauss1DMap.cpp, 884
- Gauss1DMap.h, 885
- Gauss1DMap
  - mappel::Gauss1DMap, 135
- Gauss1DMLE.cpp, 886
- Gauss1DMLE.h, 886
- Gauss1DMLE
  - mappel::Gauss1DMLE, 163
- Gauss1DModel
  - mappel::Gauss1DModel, 190
- Gauss1DModel.cpp, 887
- Gauss1DModel.h, 887
- Gauss1DSumModelT
  - mappel::Gauss2DMap, 300
  - mappel::Gauss2DMLE, 330
  - mappel::Gauss2DModel, 360
  - mappel::Gauss2DsMap, 389
- mappel::Gauss2DsMLE, 421
- mappel::Gauss2DsModel, 454
- Gauss1DsMap.cpp, 888
- Gauss1DsMap.h, 889
- Gauss1DsMap
  - mappel::Gauss1DsMap, 217, 218
- Gauss1DsMLE.cpp, 889
- Gauss1DsMLE.h, 890
- Gauss1DsMLE
  - mappel::Gauss1DsMLE, 246
- Gauss1DsModel
  - mappel::Gauss1DsModel, 274
- Gauss1DsModel.cpp, 891
- Gauss1DsModel.h, 891
- Gauss2DMap.cpp, 892
- Gauss2DMap.h, 892
- Gauss2DMap
  - mappel::Gauss2DMap, 302, 303
- Gauss2DMLE.cpp, 893
- Gauss2DMLE.h, 894
- Gauss2DMLE
  - mappel::Gauss2DMLE, 332
- Gauss2DModel
  - mappel::Gauss2DModel, 362
- Gauss2DModel.cpp, 894
- Gauss2DModel.h, 895
- Gauss2DsMap.cpp, 896
- Gauss2DsMap.h, 896
- Gauss2DsMap
  - mappel::Gauss2DsMap, 391, 392
- Gauss2DsMLE.cpp, 897
- Gauss2DsMLE.h, 898
- Gauss2DsMLE
  - mappel::Gauss2DsMLE, 424
- Gauss2DsModel
  - mappel::Gauss2DsModel, 456
- Gauss2DsModel.cpp, 898
- Gauss2DsModel.h, 899
- Gauss2DsxyMAP.h, 900
- Gauss2DsxyMAP
  - mappel::Gauss2DsxyMAP, 486
- Gauss2DsxyModel
  - mappel::Gauss2DsxyModel, 513
- Gauss2DsxyModel.h, 901
- gauss\_norm
  - mappel, 35
- gaussian\_3D\_convolution
  - mappel, 35
- gaussian\_convolution
  - mappel, 36
- generate\_poisson
  - mappel, 36
- generate\_poisson\_large
  - mappel, 36

- generate\_poisson\_small
  - mappel, 36
- get\_backtrack\_idx
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, 594
- get\_debug\_stats
  - mappel::estimator::CGaussHeuristicEstimator, 101
  - mappel::estimator::CGaussMLE, 113
  - mappel::estimator::Estimator, 125
  - mappel::estimator::HeuristicEstimator, 539
  - mappel::estimator::IterativeMaximizer, 567
  - mappel::estimator::NewtonDiagonalMaximizer, 710
  - mappel::estimator::NewtonMaximizer, 731
  - mappel::estimator::QuasiNewtonMaximizer, 795
  - mappel::estimator::SimulatedAnnealingMaximizer, 813
  - mappel::estimator::ThreadedEstimator, 849
  - mappel::estimator::TrustRegionMaximizer, 865
- get\_exit\_counts
  - mappel::estimator::CGaussHeuristicEstimator, 101
  - mappel::estimator::CGaussMLE, 114
  - mappel::estimator::Estimator, 126
  - mappel::estimator::HeuristicEstimator, 540
  - mappel::estimator::IterativeMaximizer, 568
  - mappel::estimator::NewtonDiagonalMaximizer, 711
  - mappel::estimator::NewtonMaximizer, 732
  - mappel::estimator::QuasiNewtonMaximizer, 796
  - mappel::estimator::SimulatedAnnealingMaximizer, 814
  - mappel::estimator::ThreadedEstimator, 850
  - mappel::estimator::TrustRegionMaximizer, 866
- get\_hyperparam\_index
  - mappel::Gauss1DMAP, 137
  - mappel::Gauss1DMLE, 165
  - mappel::Gauss1DModel, 192
  - mappel::Gauss1DsMAP, 219
  - mappel::Gauss1DsMLE, 248
  - mappel::Gauss1DsModel, 276
  - mappel::Gauss2DMAP, 305
  - mappel::Gauss2DMLE, 335
  - mappel::Gauss2DModel, 364
  - mappel::Gauss2DsMAP, 394
  - mappel::Gauss2DsMLE, 427
  - mappel::Gauss2DsModel, 458
  - mappel::Gauss2DsxyMAP, 488
  - mappel::Gauss2DsxyModel, 515
  - mappel::MAPEstimator, 581
  - mappel::MCMCAdaptor1Ds, 624
  - mappel::MCMCAdaptor1D, 606
  - mappel::MCMCAdaptor2Ds, 660
  - mappel::MCMCAdaptor2D, 642
  - mappel::MLEstimator, 687
  - mappel::PointEmitterModel, 750
- get\_hyperparam\_names
  - mappel::Gauss1DMAP, 137
  - mappel::Gauss1DMLE, 165
  - mappel::Gauss1DModel, 192
  - mappel::Gauss1DsMAP, 220
  - mappel::Gauss1DsMLE, 248
  - mappel::Gauss1DsModel, 276
  - mappel::Gauss2DMAP, 305
  - mappel::Gauss2DMLE, 335
  - mappel::Gauss2DModel, 364
  - mappel::Gauss2DsMAP, 395
  - mappel::Gauss2DsMLE, 427
  - mappel::Gauss2DsModel, 459
  - mappel::Gauss2DsxyMAP, 488
  - mappel::Gauss2DsxyModel, 515
  - mappel::MAPEstimator, 582
  - mappel::MCMCAdaptor1Ds, 624
  - mappel::MCMCAdaptor1D, 606
  - mappel::MCMCAdaptor2Ds, 661
  - mappel::MCMCAdaptor2D, 642
  - mappel::MLEstimator, 688
  - mappel::PointEmitterModel, 751
- get\_hyperparam\_value
  - mappel::Gauss1DMAP, 137
  - mappel::Gauss1DMLE, 165
  - mappel::Gauss1DModel, 192
  - mappel::Gauss1DsMAP, 220
  - mappel::Gauss1DsMLE, 248
  - mappel::Gauss1DsModel, 276
  - mappel::Gauss2DMAP, 305
  - mappel::Gauss2DMLE, 335
  - mappel::Gauss2DModel, 364
  - mappel::Gauss2DsMAP, 395
  - mappel::Gauss2DsMLE, 427
  - mappel::Gauss2DsModel, 459
  - mappel::Gauss2DsxyMAP, 488
  - mappel::Gauss2DsxyModel, 515
  - mappel::MAPEstimator, 582
  - mappel::MCMCAdaptor1Ds, 624
  - mappel::MCMCAdaptor1D, 606
  - mappel::MCMCAdaptor2Ds, 661
  - mappel::MCMCAdaptor2D, 642
  - mappel::MLEstimator, 688
  - mappel::PointEmitterModel, 751
- get\_hyperparams
  - mappel::Gauss1DMAP, 137
  - mappel::Gauss1DMLE, 165
  - mappel::Gauss1DModel, 192
  - mappel::Gauss1DsMAP, 220
  - mappel::Gauss1DsMLE, 248
  - mappel::Gauss1DsModel, 276
  - mappel::Gauss2DMAP, 305
  - mappel::Gauss2DMLE, 335
  - mappel::Gauss2DModel, 364
  - mappel::Gauss2DsMAP, 395

- mappel::Gauss2DsMLE, [427](#)
- mappel::Gauss2DsModel, [459](#)
- mappel::Gauss2DsxyMAP, [488](#)
- mappel::Gauss2DsxyModel, [515](#)
- mappel::MAPEstimator, [582](#)
- mappel::MCMCAdaptor1Ds, [624](#)
- mappel::MCMCAdaptor1D, [607](#)
- mappel::MCMCAdaptor2Ds, [661](#)
- mappel::MCMCAdaptor2D, [642](#)
- mappel::MLEstimator, [688](#)
- mappel::PointEmitterModel, [751](#)
- get\_image\_from\_stack
  - mappel::Gauss1DMAP, [137](#)
  - mappel::Gauss1DMLE, [165](#)
  - mappel::Gauss1DModel, [193](#)
  - mappel::Gauss1DsMAP, [220](#)
  - mappel::Gauss1DsMLE, [248](#)
  - mappel::Gauss1DsModel, [276](#)
  - mappel::Gauss2DMAP, [305](#)
  - mappel::Gauss2DMLE, [335](#)
  - mappel::Gauss2DModel, [365](#)
  - mappel::Gauss2DsMAP, [395](#)
  - mappel::Gauss2DsMLE, [427](#)
  - mappel::Gauss2DsModel, [459](#)
  - mappel::Gauss2DsxyMAP, [489](#)
  - mappel::Gauss2DsxyModel, [515](#)
  - mappel::ImageFormat1DBase, [546](#)
  - mappel::ImageFormat2DBase, [553](#)
  - mappel::PoissonNoise1DObjective, [767](#)
  - mappel::PoissonNoise2DObjective, [774](#)
- get\_lbound
  - mappel::Gauss1DMAP, [137](#)
  - mappel::Gauss1DMLE, [165](#)
  - mappel::Gauss1DModel, [193](#)
  - mappel::Gauss1DsMAP, [220](#)
  - mappel::Gauss1DsMLE, [248](#)
  - mappel::Gauss1DsModel, [276](#)
  - mappel::Gauss2DMAP, [305](#)
  - mappel::Gauss2DMLE, [335](#)
  - mappel::Gauss2DModel, [365](#)
  - mappel::Gauss2DsMAP, [395](#)
  - mappel::Gauss2DsMLE, [427](#)
  - mappel::Gauss2DsModel, [459](#)
  - mappel::Gauss2DsxyMAP, [489](#)
  - mappel::Gauss2DsxyModel, [515](#)
  - mappel::MAPEstimator, [582](#)
  - mappel::MCMCAdaptor1Ds, [625](#)
  - mappel::MCMCAdaptor1D, [607](#)
  - mappel::MCMCAdaptor2Ds, [661](#)
  - mappel::MCMCAdaptor2D, [643](#)
  - mappel::MLEstimator, [688](#)
  - mappel::PointEmitterModel, [751](#)
- get\_max\_sigma
  - mappel::Gauss1DsMAP, [220](#)
- mappel::Gauss1DsMLE, [249](#)
- mappel::Gauss1DsModel, [277](#)
- mappel::Gauss2DsMAP, [395](#)
- mappel::Gauss2DsMLE, [427](#), [428](#)
- mappel::Gauss2DsModel, [459](#)
- mappel::Gauss2DsxyModel, [515](#), [516](#)
- get\_max\_sigma\_ratio
  - mappel::Gauss2DsMAP, [396](#)
  - mappel::Gauss2DsMLE, [428](#)
  - mappel::Gauss2DsModel, [460](#)
  - mappel::Gauss2DsxyModel, [516](#)
- get\_mcmc\_num\_phases
  - mappel::Gauss1DMAP, [138](#)
  - mappel::Gauss1DMLE, [166](#)
  - mappel::Gauss1DModel, [193](#)
  - mappel::Gauss1DsMAP, [220](#)
  - mappel::Gauss1DsMLE, [249](#)
  - mappel::Gauss1DsModel, [277](#)
  - mappel::Gauss2DMAP, [305](#)
  - mappel::Gauss2DMLE, [335](#)
  - mappel::Gauss2DModel, [365](#)
  - mappel::Gauss2DsMAP, [396](#)
  - mappel::Gauss2DsMLE, [428](#)
  - mappel::Gauss2DsModel, [460](#)
  - mappel::Gauss2DsxyMAP, [489](#)
  - mappel::MCMCAdaptor1Ds, [625](#)
  - mappel::MCMCAdaptor1D, [607](#)
  - mappel::MCMCAdaptor2Ds, [661](#)
  - mappel::MCMCAdaptor2D, [643](#)
  - mappel::MCMCAdaptorBase, [675](#)
- get\_mcmc\_sigma\_scale
  - mappel::Gauss1DMAP, [138](#)
  - mappel::Gauss1DMLE, [166](#)
  - mappel::Gauss1DModel, [193](#)
  - mappel::Gauss1DsMAP, [221](#)
  - mappel::Gauss1DsMLE, [249](#)
  - mappel::Gauss1DsModel, [277](#)
  - mappel::Gauss2DMAP, [306](#)
  - mappel::Gauss2DMLE, [336](#)
  - mappel::Gauss2DModel, [365](#)
  - mappel::Gauss2DsMAP, [396](#)
  - mappel::Gauss2DsMLE, [428](#)
  - mappel::Gauss2DsModel, [460](#)
  - mappel::Gauss2DsxyMAP, [489](#)
  - mappel::MCMCAdaptor1Ds, [625](#)
  - mappel::MCMCAdaptor1D, [607](#)
  - mappel::MCMCAdaptor2Ds, [661](#)
  - mappel::MCMCAdaptor2D, [643](#)
  - mappel::MCMCAdaptorBase, [675](#)
- get\_min\_sigma
  - mappel::Gauss1DsMAP, [221](#)
  - mappel::Gauss1DsMLE, [249](#)
  - mappel::Gauss1DsModel, [277](#)
  - mappel::Gauss2DsMAP, [396](#)



- mappel::Gauss2DsMLE, 428
- mappel::Gauss2DsModel, 460
- mappel::Gauss2DsxyModel, 516
- get\_model
  - mappel::estimator::CGaussHeuristicEstimator, 102
  - mappel::estimator::CGaussMLE, 114
  - mappel::estimator::Estimator, 126
  - mappel::estimator::HeuristicEstimator, 540
  - mappel::estimator::IterativeMaximizer, 568
  - mappel::estimator::NewtonDiagonalMaximizer, 711
  - mappel::estimator::NewtonMaximizer, 732
  - mappel::estimator::QuasiNewtonMaximizer, 796
  - mappel::estimator::SimulatedAnnealingMaximizer, 814
  - mappel::estimator::ThreadedEstimator, 850
  - mappel::estimator::TrustRegionMaximizer, 866
- get\_num\_hyperparams
  - mappel::Gauss1DMAP, 138
  - mappel::Gauss1DMLE, 166
  - mappel::Gauss1DModel, 193
  - mappel::Gauss1DsMAP, 221
  - mappel::Gauss1DsMLE, 249
  - mappel::Gauss1DsModel, 277
  - mappel::Gauss2DMAP, 306
  - mappel::Gauss2DMLE, 336
  - mappel::Gauss2DModel, 365
  - mappel::Gauss2DsMAP, 396
  - mappel::Gauss2DsMLE, 428
  - mappel::Gauss2DsModel, 460
  - mappel::Gauss2DsxyMAP, 489
  - mappel::Gauss2DsxyModel, 516
  - mappel::MAPEstimator, 582
  - mappel::MCMCAdaptor1Ds, 625
  - mappel::MCMCAdaptor1D, 607
  - mappel::MCMCAdaptor2Ds, 661
  - mappel::MCMCAdaptor2D, 643
  - mappel::MLEstimator, 688
  - mappel::PointEmitterModel, 751
- get\_num\_params
  - mappel::Gauss1DMAP, 138
  - mappel::Gauss1DMLE, 166
  - mappel::Gauss1DModel, 193
  - mappel::Gauss1DsMAP, 221
  - mappel::Gauss1DsMLE, 249
  - mappel::Gauss1DsModel, 277
  - mappel::Gauss2DMAP, 306
  - mappel::Gauss2DMLE, 336
  - mappel::Gauss2DModel, 365
  - mappel::Gauss2DsMAP, 396
  - mappel::Gauss2DsMLE, 429
  - mappel::Gauss2DsModel, 460
  - mappel::Gauss2DsxyMAP, 489
  - mappel::Gauss2DsxyModel, 516
  - mappel::MAPEstimator, 582
  - mappel::MCMCAdaptor1Ds, 625
  - mappel::MCMCAdaptor1D, 607
  - mappel::MCMCAdaptor2Ds, 662
  - mappel::MCMCAdaptor2D, 643
  - mappel::MLEstimator, 688
  - mappel::PointEmitterModel, 751
- mappel::MCMCAdaptor1Ds, 625
- mappel::MCMCAdaptor1D, 607
- mappel::MCMCAdaptor2Ds, 661
- mappel::MCMCAdaptor2D, 643
- mappel::MLEstimator, 688
- mappel::PointEmitterModel, 751
- get\_num\_pixels
  - mappel::Gauss1DMAP, 138
  - mappel::Gauss1DMLE, 166
  - mappel::Gauss1DModel, 193
  - mappel::Gauss1DsMAP, 221
  - mappel::Gauss1DsMLE, 249
  - mappel::Gauss1DsModel, 277
  - mappel::Gauss2DMAP, 306
  - mappel::Gauss2DMLE, 336
  - mappel::Gauss2DModel, 365
  - mappel::Gauss2DsMAP, 397
  - mappel::Gauss2DsMLE, 429
  - mappel::Gauss2DsModel, 460
  - mappel::Gauss2DsxyMAP, 489
  - mappel::Gauss2DsxyModel, 516
  - mappel::ImageFormat1DBase, 547
  - mappel::ImageFormat2DBase, 553
  - mappel::PoissonNoise1DObjective, 767
  - mappel::PoissonNoise2DObjective, 774
- get\_param\_names
  - mappel::Gauss1DMAP, 138
  - mappel::Gauss1DMLE, 166
  - mappel::Gauss1DModel, 194
  - mappel::Gauss1DsMAP, 221
  - mappel::Gauss1DsMLE, 250
  - mappel::Gauss1DsModel, 278
  - mappel::Gauss2DMAP, 306
  - mappel::Gauss2DMLE, 336
  - mappel::Gauss2DModel, 366
  - mappel::Gauss2DsMAP, 397
  - mappel::Gauss2DsMLE, 429
  - mappel::Gauss2DsModel, 461
  - mappel::Gauss2DsxyMAP, 490
  - mappel::Gauss2DsxyModel, 516
  - mappel::MAPEstimator, 582
  - mappel::MCMCAdaptor1Ds, 625
  - mappel::MCMCAdaptor1D, 607
  - mappel::MCMCAdaptor2Ds, 662
  - mappel::MCMCAdaptor2D, 643
  - mappel::MLEstimator, 688
  - mappel::PointEmitterModel, 751
- get\_prior
  - mappel::Gauss1DMAP, 138, 139
  - mappel::Gauss1DMLE, 166, 167
  - mappel::Gauss1DModel, 194
  - mappel::Gauss1DsMAP, 221, 222
  - mappel::Gauss1DsMLE, 250
  - mappel::Gauss1DsModel, 278

- mappel::Gauss2DModel, 366
- mappel::Gauss2DsMAP, 397
- mappel::Gauss2DsMLE, 429
- mappel::Gauss2DsModel, 461
- mappel::Gauss2DsxyMAP, 490
- mappel::Gauss2DsxyModel, 517
- mappel::MAPEstimator, 583
- mappel::MCMCAdaptor1Ds, 625, 626
- mappel::MCMCAdaptor1D, 608
- mappel::MCMCAdaptor2Ds, 662
- mappel::MCMCAdaptor2D, 643, 644
- mappel::MLEstimator, 689
- mappel::PointEmitterModel, 752
- get\_psf\_sigma
  - mappel::Gauss1DModel, 194
  - mappel::Gauss2DModel, 366
  - mappel::Gauss2DsxyMAP, 490
- get\_rng\_generator
  - mappel::Gauss1DModel, 194
  - mappel::Gauss1DsMAP, 222
  - mappel::Gauss1DsMLE, 250
  - mappel::Gauss1DsModel, 278
  - mappel::Gauss2DModel, 366
  - mappel::Gauss2DsMAP, 397
  - mappel::Gauss2DsMLE, 429
  - mappel::Gauss2DsModel, 461
  - mappel::Gauss2DsxyMAP, 490
  - mappel::Gauss2DsxyModel, 517
  - mappel::MAPEstimator, 583
  - mappel::MCMCAdaptor1Ds, 626
  - mappel::MCMCAdaptor1D, 608
  - mappel::MCMCAdaptor2Ds, 662
  - mappel::MCMCAdaptor2D, 644
  - mappel::MLEstimator, 689
  - mappel::PointEmitterModel, 752
- get\_rng\_manager
  - mappel::Gauss1DModel, 194
  - mappel::Gauss1DsMAP, 222
  - mappel::Gauss1DsMLE, 250
  - mappel::Gauss1DsModel, 278
  - mappel::Gauss2DModel, 366
  - mappel::Gauss2DsMLE, 429
- mappel::Gauss2DModel, 366
- mappel::Gauss2DsMAP, 397
- mappel::Gauss2DsMLE, 429
- mappel::Gauss2DsModel, 461
- mappel::Gauss2DsxyMAP, 490
- mappel::Gauss2DsxyModel, 517
- mappel::MAPEstimator, 583
- mappel::MCMCAdaptor1Ds, 626
- mappel::MCMCAdaptor1D, 608
- mappel::MCMCAdaptor2Ds, 662
- mappel::MCMCAdaptor2D, 644
- mappel::MLEstimator, 689
- mappel::PointEmitterModel, 752
- get\_sequence\_len
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, 594
- get\_size
  - mappel::Gauss1DModel, 195
  - mappel::Gauss1DsMAP, 222
  - mappel::Gauss1DsMLE, 250
  - mappel::Gauss1DsModel, 278
  - mappel::Gauss2DModel, 367
  - mappel::Gauss2DsMAP, 397, 398
  - mappel::Gauss2DsMLE, 430
  - mappel::Gauss2DsModel, 461
  - mappel::Gauss2DsxyMAP, 491
  - mappel::Gauss2DsxyModel, 517
  - mappel::ImageFormat1DBase, 547
  - mappel::ImageFormat2DBase, 553
  - mappel::PoissonNoise1DObjective, 767
  - mappel::PoissonNoise2DObjective, 774
- get\_size\_image\_stack
  - mappel::Gauss1DModel, 195
  - mappel::Gauss1DsMAP, 222
  - mappel::Gauss1DsMLE, 251
  - mappel::Gauss1DsModel, 279
  - mappel::Gauss2DModel, 367
  - mappel::Gauss2DsMAP, 398
  - mappel::Gauss2DsMLE, 430
  - mappel::Gauss2DsModel, 462
  - mappel::Gauss2DsxyMAP, 491
  - mappel::Gauss2DsxyModel, 517
  - mappel::ImageFormat1DBase, 547
  - mappel::ImageFormat2DBase, 554
  - mappel::PoissonNoise1DObjective, 767
  - mappel::PoissonNoise2DObjective, 774



- get\_stats
  - mappel::Gauss1DMAP, 140
  - mappel::Gauss1DMLE, 168
  - mappel::Gauss1DModel, 195
  - mappel::Gauss1DsMAP, 222
  - mappel::Gauss1DsMLE, 251
  - mappel::Gauss1DsModel, 279
  - mappel::Gauss2DMAP, 308
  - mappel::Gauss2DMLE, 338
  - mappel::Gauss2DModel, 367
  - mappel::Gauss2DsMAP, 398
  - mappel::Gauss2DsMLE, 430
  - mappel::Gauss2DsModel, 462
  - mappel::Gauss2DsxyMAP, 491
  - mappel::Gauss2DsxyModel, 518
  - mappel::ImageFormat1DBase, 547
  - mappel::ImageFormat2DBase, 554
  - mappel::MAPEstimator, 583
  - mappel::MCMCAdaptor1Ds, 626
  - mappel::MCMCAdaptor1D, 608
  - mappel::MCMCAdaptor2Ds, 662
  - mappel::MCMCAdaptor2D, 644
  - mappel::MCMCAdaptorBase, 676
  - mappel::MLEstimator, 689
  - mappel::PointEmitterModel, 752
  - mappel::PoissonNoise1DObjective, 767
  - mappel::PoissonNoise2DObjective, 774
  - mappel::estimator::CGaussHeuristicEstimator, 102
  - mappel::estimator::CGaussMLE, 114
  - mappel::estimator::Estimator, 126
  - mappel::estimator::HeuristicEstimator, 540
  - mappel::estimator::IterativeMaximizer, 568
  - mappel::estimator::NewtonDiagonalMaximizer, 711
  - mappel::estimator::NewtonMaximizer, 732
  - mappel::estimator::QuasiNewtonMaximizer, 796
  - mappel::estimator::SimulatedAnnealingMaximizer, 814
  - mappel::estimator::ThreadedEstimator, 850
  - mappel::estimator::TrustRegionMaximizer, 866
- get\_theta\_sequence
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 594
- get\_theta\_sequence\_rllh
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 595
- get\_total\_backtracks
  - mappel::estimator::IterativeMaximizer, 568
  - mappel::estimator::NewtonDiagonalMaximizer, 711
  - mappel::estimator::NewtonMaximizer, 732
  - mappel::estimator::QuasiNewtonMaximizer, 796
  - mappel::estimator::TrustRegionMaximizer, 866
- get\_total\_der\_evals
  - mappel::estimator::IterativeMaximizer, 569
  - mappel::estimator::NewtonDiagonalMaximizer, 712
  - mappel::estimator::NewtonMaximizer, 733
  - mappel::estimator::QuasiNewtonMaximizer, 797
  - mappel::estimator::TrustRegionMaximizer, 867
- get\_total\_fun\_evals
  - mappel::estimator::IterativeMaximizer, 569
  - mappel::estimator::NewtonDiagonalMaximizer, 712
  - mappel::estimator::NewtonMaximizer, 733
  - mappel::estimator::QuasiNewtonMaximizer, 797
  - mappel::estimator::TrustRegionMaximizer, 867
- get\_total\_iterations
  - mappel::estimator::IterativeMaximizer, 569
  - mappel::estimator::NewtonDiagonalMaximizer, 712
  - mappel::estimator::NewtonMaximizer, 733
  - mappel::estimator::QuasiNewtonMaximizer, 797
  - mappel::estimator::TrustRegionMaximizer, 867
- get\_ubound
  - mappel::Gauss1DMAP, 140
  - mappel::Gauss1DMLE, 168
  - mappel::Gauss1DModel, 195
  - mappel::Gauss1DsMAP, 223
  - mappel::Gauss1DsMLE, 251
  - mappel::Gauss1DsModel, 279
  - mappel::Gauss2DMAP, 308
  - mappel::Gauss2DMLE, 338
  - mappel::Gauss2DModel, 367
  - mappel::Gauss2DsMAP, 398
  - mappel::Gauss2DsMLE, 430
  - mappel::Gauss2DsModel, 462
  - mappel::Gauss2DsxyMAP, 491
  - mappel::Gauss2DsxyModel, 518
  - mappel::MAPEstimator, 583
  - mappel::MCMCAdaptor1Ds, 626
  - mappel::MCMCAdaptor1D, 608
  - mappel::MCMCAdaptor2Ds, 662
  - mappel::MCMCAdaptor2D, 644
  - mappel::MLEstimator, 689
  - mappel::PointEmitterModel, 752
- global\_default\_mcmc\_sigma\_scale
  - mappel::Gauss1DMAP, 152
  - mappel::Gauss1DMLE, 180
  - mappel::Gauss1DModel, 207
  - mappel::Gauss1DsMAP, 236
  - mappel::Gauss1DsMLE, 264
  - mappel::Gauss1DsModel, 292
  - mappel::Gauss2DMAP, 321
  - mappel::Gauss2DMLE, 351
  - mappel::Gauss2DModel, 380
  - mappel::Gauss2DsMAP, 412
  - mappel::Gauss2DsMLE, 444
  - mappel::Gauss2DsModel, 476
  - mappel::Gauss2DsxyMAP, 503
  - mappel::MCMCAdaptor1Ds, 635
  - mappel::MCMCAdaptor1D, 617
  - mappel::MCMCAdaptor2Ds, 672

- mappel::MCMCAdaptor2D, 653
- mappel::MCMCAdaptorBase, 676
- global\_max\_mcmc\_sigma\_scale
  - mappel::Gauss1DMAP, 153
  - mappel::Gauss1DMLE, 181
  - mappel::Gauss1DModel, 207
  - mappel::Gauss1DsMAP, 236
  - mappel::Gauss1DsMLE, 264
  - mappel::Gauss1DsModel, 292
  - mappel::Gauss2DMAP, 321
  - mappel::Gauss2DMLE, 351
  - mappel::Gauss2DModel, 380
  - mappel::Gauss2DsMAP, 412
  - mappel::Gauss2DsMLE, 445
  - mappel::Gauss2DsModel, 476
  - mappel::Gauss2DsxyMAP, 503
  - mappel::MCMCAdaptor1Ds, 635
  - mappel::MCMCAdaptor1D, 617
  - mappel::MCMCAdaptor2Ds, 672
  - mappel::MCMCAdaptor2D, 653
  - mappel::MCMCAdaptorBase, 676
- global\_max\_psf\_sigma
  - mappel::Gauss1DMAP, 153
  - mappel::Gauss1DMLE, 181
  - mappel::Gauss1DModel, 207
  - mappel::Gauss1DsMAP, 236
  - mappel::Gauss1DsMLE, 264
  - mappel::Gauss1DsModel, 292
  - mappel::Gauss2DMAP, 321
  - mappel::Gauss2DMLE, 351
  - mappel::Gauss2DModel, 380
  - mappel::Gauss2DsMAP, 413
  - mappel::Gauss2DsMLE, 445
  - mappel::Gauss2DsModel, 476
  - mappel::Gauss2DsxyMAP, 504
  - mappel::Gauss2DsxyModel, 527
  - mappel::MAPEstimator, 590
  - mappel::MCMCAdaptor1Ds, 635
  - mappel::MCMCAdaptor1D, 617
  - mappel::MCMCAdaptor2Ds, 672
  - mappel::MCMCAdaptor2D, 653
  - mappel::MLEstimator, 696
  - mappel::PointEmitterModel, 759
- global\_min\_psf\_sigma
  - mappel::Gauss1DMAP, 153
  - mappel::Gauss1DMLE, 181
  - mappel::Gauss1DModel, 208
  - mappel::Gauss1DsMAP, 236
  - mappel::Gauss1DsMLE, 265
  - mappel::Gauss1DsModel, 292
  - mappel::Gauss2DMAP, 321
  - mappel::Gauss2DMLE, 351
  - mappel::Gauss2DModel, 381
  - mappel::Gauss2DsMAP, 413
  - mappel::Gauss2DsMLE, 445
  - mappel::Gauss2DsModel, 476
  - mappel::Gauss2DsxyMAP, 504
  - mappel::Gauss2DsxyModel, 527
  - mappel::MAPEstimator, 590
  - mappel::MCMCAdaptor1Ds, 635
  - mappel::MCMCAdaptor1D, 617
  - mappel::MCMCAdaptor2Ds, 672
  - mappel::MCMCAdaptor2D, 653
  - mappel::MLEstimator, 696
  - mappel::PointEmitterModel, 759
- global\_min\_size
  - mappel::Gauss1DMAP, 153
  - mappel::Gauss1DMLE, 181
  - mappel::Gauss1DModel, 208
  - mappel::Gauss1DsMAP, 237
  - mappel::Gauss1DsMLE, 265
  - mappel::Gauss1DsModel, 293
  - mappel::Gauss2DMAP, 322
  - mappel::Gauss2DMLE, 352
  - mappel::Gauss2DModel, 381
  - mappel::Gauss2DsMAP, 413
  - mappel::Gauss2DsMLE, 445
  - mappel::Gauss2DsModel, 476
  - mappel::Gauss2DsxyMAP, 504
  - mappel::Gauss2DsxyModel, 527
  - mappel::ImageFormat1DBase, 548
  - mappel::ImageFormat2DBase, 555
  - mappel::PoissonNoise1DObjective, 769
  - mappel::PoissonNoise2DObjective, 776
- grad
  - mappel::estimator::IterativeMaximizer::Maximizer←  
Data, 599
  - mappel::methods::likelihood, 69
  - mappel::methods::objective, 74

- grad2
  - mappel::methods::likelihood, [69](#)
  - mappel::methods::objective, [74](#), [75](#)
- grad\_components
  - mappel::methods::likelihood::debug, [71](#)
  - mappel::methods::objective::debug, [78](#), [79](#)
- grad\_stack
  - mappel::methods::objective::openmp, [81](#)
- GradRatio
  - mappel::estimator, [50](#)
- Gx
  - mappel::Gauss1DModel::Stencil, [839](#)
  - mappel::Gauss1DsModel::Stencil, [825](#)
  - mappel::Gauss2DModel::Stencil, [830](#)
  - mappel::Gauss2DsModel::Stencil, [835](#)
  - mappel::Gauss2DsxyModel::Stencil, [821](#)
- Gy
  - mappel::Gauss2DModel::Stencil, [830](#)
  - mappel::Gauss2DsModel::Stencil, [836](#)
  - mappel::Gauss2DsxyModel::Stencil, [821](#)
- has\_fixed\_parameters
  - mappel::estimator::IterativeMaximizer::Maximizer↔  
Data, [595](#)
- has\_hyperparam
  - mappel::Gauss1DMAP, [140](#)
  - mappel::Gauss1DMLE, [168](#)
  - mappel::Gauss1DModel, [195](#)
  - mappel::Gauss1DsMAP, [223](#)
  - mappel::Gauss1DsMLE, [251](#)
  - mappel::Gauss1DsModel, [279](#)
  - mappel::Gauss2DMAP, [308](#)
  - mappel::Gauss2DMLE, [338](#)
  - mappel::Gauss2DModel, [367](#)
  - mappel::Gauss2DsMAP, [398](#)
  - mappel::Gauss2DsMLE, [430](#)
  - mappel::Gauss2DsModel, [462](#)
  - mappel::Gauss2DsxyMAP, [491](#)
  - mappel::Gauss2DsxyModel, [518](#)
  - mappel::MAPEstimator, [583](#)
  - mappel::MCMCAdaptor1Ds, [626](#)
  - mappel::MCMCAdaptor1D, [608](#)
  - mappel::MCMCAdaptor2Ds, [663](#)
  - mappel::MCMCAdaptor2D, [644](#)
  - mappel::MLEstimator, [689](#)
  - mappel::PointEmitterModel, [752](#)
- has\_theta\_sequence
  - mappel::estimator::IterativeMaximizer::Maximizer↔  
Data, [595](#)
- hessian
  - mappel::methods::likelihood, [69](#)
  - mappel::methods::objective, [75](#), [76](#)
- hessian\_components
  - mappel::methods::likelihood::debug, [71](#)
- mappel::methods::objective::debug, [79](#)
- hessian\_stack
  - mappel::methods::objective::openmp, [81](#)
- HeuristicEstimator
  - mappel::estimator::HeuristicEstimator, [532](#)
- I
  - mappel::Gauss1DModel::Stencil, [838](#)
  - mappel::Gauss1DsModel::Stencil, [824](#)
  - mappel::Gauss2DModel::Stencil, [828](#)
  - mappel::Gauss2DsModel::Stencil, [833](#)
  - mappel::Gauss2DsxyModel::Stencil, [819](#)
- icontains
  - mappel, [36](#)
- IdxMatT
  - mappel, [27](#)
- IdxVecT
  - mappel, [27](#)
- IdxT
  - mappel, [27](#)
- im
  - mappel::estimator::IterativeMaximizer::Maximizer↔  
Data, [599](#)
- ImageCoordT
  - mappel, [27](#)
  - mappel::Gauss1DMAP, [133](#)
  - mappel::Gauss1DMLE, [161](#)
  - mappel::Gauss1DModel, [188](#)
  - mappel::Gauss1DsMAP, [215](#)
  - mappel::Gauss1DsMLE, [244](#)
  - mappel::Gauss1DsModel, [272](#)
  - mappel::Gauss2DMAP, [300](#)
  - mappel::Gauss2DMLE, [330](#)
  - mappel::Gauss2DModel, [360](#)
  - mappel::Gauss2DsMAP, [389](#)
  - mappel::Gauss2DsMLE, [421](#)
  - mappel::Gauss2DsModel, [454](#)
  - mappel::Gauss2DsxyMAP, [484](#)
  - mappel::Gauss2DsxyModel, [511](#)
  - mappel::ImageFormat1DBase, [545](#)
  - mappel::ImageFormat2DBase, [551](#)
  - mappel::PoissonNoise1DObjective, [764](#)
  - mappel::PoissonNoise2DObjective, [771](#)
- ImageFormat1DBase
  - mappel::ImageFormat1DBase, [546](#)
- ImageFormat1DBase.cpp, [902](#)
- ImageFormat1DBase.h, [902](#)
- ImageFormat2DBase
  - mappel::ImageFormat2DBase, [552](#)
- ImageFormat2DBase.cpp, [903](#)
- ImageFormat2DBase.h, [904](#)
- ImagePixelT
  - mappel, [27](#)
  - mappel::Gauss1DMAP, [133](#)

- mappel::Gauss1DMLE, [161](#)
- mappel::Gauss1DModel, [188](#)
- mappel::Gauss1DsMAP, [215](#)
- mappel::Gauss1DsMLE, [244](#)
- mappel::Gauss1DsModel, [272](#)
- mappel::Gauss2DMAP, [300](#)
- mappel::Gauss2DMLE, [330](#)
- mappel::Gauss2DModel, [360](#)
- mappel::Gauss2DsMAP, [389](#)
- mappel::Gauss2DsMLE, [422](#)
- mappel::Gauss2DsModel, [454](#)
- mappel::Gauss2DsxyMAP, [484](#)
- mappel::Gauss2DsxyModel, [511](#)
- mappel::ImageFormat1DBase, [545](#)
- mappel::ImageFormat2DBase, [551](#)
- mappel::PoissonNoise1DObjective, [764](#)
- mappel::PoissonNoise2DObjective, [771](#)
- ImageShapeT
  - mappel::Gauss1DMAP, [133](#)
  - mappel::Gauss1DMLE, [161](#)
  - mappel::Gauss1DModel, [188](#)
  - mappel::Gauss1DsMAP, [215](#)
  - mappel::Gauss1DsMLE, [244](#)
  - mappel::Gauss1DsModel, [272](#)
  - mappel::Gauss2DMAP, [300](#)
  - mappel::Gauss2DMLE, [330](#)
  - mappel::Gauss2DModel, [360](#)
  - mappel::Gauss2DsMAP, [389](#)
  - mappel::Gauss2DsMLE, [422](#)
  - mappel::Gauss2DsModel, [454](#)
  - mappel::Gauss2DsxyMAP, [484](#)
  - mappel::Gauss2DsxyModel, [511](#)
  - mappel::ImageFormat1DBase, [545](#)
  - mappel::ImageFormat2DBase, [551](#)
  - mappel::PoissonNoise1DObjective, [764](#)
  - mappel::PoissonNoise2DObjective, [771](#)
- ImageSizeShapeT
  - mappel::Gauss1DMAP, [133](#)
  - mappel::Gauss1DMLE, [161](#)
  - mappel::Gauss1DModel, [189](#)
  - mappel::Gauss1DsMAP, [215](#)
  - mappel::Gauss1DsMLE, [244](#)
  - mappel::Gauss1DsModel, [272](#)
  - mappel::Gauss2DMAP, [300](#)
  - mappel::Gauss2DMLE, [330](#)
  - mappel::Gauss2DModel, [360](#)
  - mappel::Gauss2DsMAP, [390](#)
  - mappel::Gauss2DsMLE, [422](#)
  - mappel::Gauss2DsModel, [454](#)
  - mappel::Gauss2DsxyMAP, [484](#)
  - mappel::Gauss2DsxyModel, [511](#)
  - mappel::ImageFormat1DBase, [545](#)
  - mappel::ImageFormat2DBase, [551](#)
  - mappel::PoissonNoise1DObjective, [765](#)
- mappel::PoissonNoise2DObjective, [772](#)
- ImageSizeVecShapeT
  - mappel::Gauss1DMAP, [133](#)
  - mappel::Gauss1DMLE, [161](#)
  - mappel::Gauss1DModel, [189](#)
  - mappel::Gauss1DsMAP, [216](#)
  - mappel::Gauss1DsMLE, [244](#)
  - mappel::Gauss1DsModel, [273](#)
  - mappel::Gauss2DMAP, [300](#)
  - mappel::Gauss2DMLE, [330](#)
  - mappel::Gauss2DModel, [360](#)
  - mappel::Gauss2DsMAP, [390](#)
  - mappel::Gauss2DsMLE, [422](#)
  - mappel::Gauss2DsModel, [454](#)
  - mappel::Gauss2DsxyMAP, [485](#)
  - mappel::Gauss2DsxyModel, [511](#)
  - mappel::ImageFormat1DBase, [545](#)
  - mappel::ImageFormat2DBase, [551](#)
  - mappel::PoissonNoise1DObjective, [765](#)
  - mappel::PoissonNoise2DObjective, [772](#)
- ImageSizeVecT
  - mappel::Gauss1DMAP, [133](#)
  - mappel::Gauss1DMLE, [161](#)
  - mappel::Gauss1DModel, [189](#)
  - mappel::Gauss1DsMAP, [216](#)
  - mappel::Gauss1DsMLE, [244](#)
  - mappel::Gauss1DsModel, [273](#)
  - mappel::Gauss2DMAP, [301](#)
  - mappel::Gauss2DMLE, [331](#)
  - mappel::Gauss2DModel, [361](#)
  - mappel::Gauss2DsMAP, [390](#)
  - mappel::Gauss2DsMLE, [422](#)
  - mappel::Gauss2DsModel, [455](#)
  - mappel::Gauss2DsxyMAP, [485](#)
  - mappel::Gauss2DsxyModel, [512](#)
  - mappel::ImageFormat1DBase, [545](#)
  - mappel::ImageFormat2DBase, [551](#)
  - mappel::PoissonNoise1DObjective, [765](#)
  - mappel::PoissonNoise2DObjective, [772](#)
- ImageSizeT
  - mappel::Gauss1DMAP, [133](#)
  - mappel::Gauss1DMLE, [161](#)
  - mappel::Gauss1DModel, [189](#)
  - mappel::Gauss1DsMAP, [216](#)
  - mappel::Gauss1DsMLE, [244](#)
  - mappel::Gauss1DsModel, [272](#)
  - mappel::Gauss2DMAP, [300](#)
  - mappel::Gauss2DMLE, [330](#)
  - mappel::Gauss2DModel, [360](#)
  - mappel::Gauss2DsMAP, [390](#)
  - mappel::Gauss2DsMLE, [422](#)
  - mappel::Gauss2DsModel, [454](#)
  - mappel::Gauss2DsxyMAP, [485](#)
  - mappel::Gauss2DsxyModel, [511](#)

- mappel::ImageFormat1DBase, [545](#)
- mappel::ImageFormat2DBase, [551](#)
- mappel::PoissonNoise1DObjective, [765](#)
- mappel::PoissonNoise2DObjective, [772](#)
- ImageStackShapeT
  - mappel::Gauss1DModel, [133](#)
  - mappel::Gauss1DModel, [161](#)
  - mappel::Gauss1DModel, [189](#)
  - mappel::Gauss1DsMAP, [216](#)
  - mappel::Gauss1DsMLE, [244](#)
  - mappel::Gauss1DsModel, [273](#)
  - mappel::Gauss2DModel, [301](#)
  - mappel::Gauss2DModel, [331](#)
  - mappel::Gauss2DModel, [361](#)
  - mappel::Gauss2DsMAP, [390](#)
  - mappel::Gauss2DsMLE, [422](#)
  - mappel::Gauss2DsModel, [455](#)
  - mappel::Gauss2DsxyMAP, [485](#)
  - mappel::Gauss2DsxyModel, [512](#)
  - mappel::ImageFormat1DBase, [545](#)
  - mappel::ImageFormat2DBase, [552](#)
  - mappel::PoissonNoise1DObjective, [765](#)
  - mappel::PoissonNoise2DObjective, [772](#)
- ImageStackT
  - mappel, [27](#)
  - mappel::Gauss1DModel, [134](#)
  - mappel::Gauss1DModel, [162](#)
  - mappel::Gauss1DModel, [189](#)
  - mappel::Gauss1DsMAP, [216](#)
  - mappel::Gauss1DsMLE, [245](#)
  - mappel::Gauss1DsModel, [273](#)
  - mappel::Gauss2DModel, [301](#)
  - mappel::Gauss2DModel, [331](#)
  - mappel::Gauss2DModel, [361](#)
  - mappel::Gauss2DsMAP, [390](#)
  - mappel::Gauss2DsMLE, [423](#)
  - mappel::Gauss2DsModel, [455](#)
  - mappel::Gauss2DsxyMAP, [485](#)
  - mappel::Gauss2DsxyModel, [512](#)
  - mappel::ImageFormat1DBase, [545](#)
  - mappel::ImageFormat2DBase, [552](#)
  - mappel::PoissonNoise1DObjective, [765](#)
  - mappel::PoissonNoise2DObjective, [772](#)
- ImageT
  - mappel, [27](#)
  - mappel::Gauss1DModel, [134](#)
  - mappel::Gauss1DModel, [162](#)
  - mappel::Gauss1DModel, [189](#)
  - mappel::Gauss1DsMAP, [216](#)
  - mappel::Gauss1DsMLE, [245](#)
  - mappel::Gauss1DsModel, [273](#)
  - mappel::Gauss2DModel, [301](#)
  - mappel::Gauss2DModel, [331](#)
  - mappel::Gauss2DModel, [361](#)
- mappel::Gauss2DsMAP, [390](#)
- mappel::Gauss2DsMLE, [423](#)
- mappel::Gauss2DsModel, [455](#)
- mappel::Gauss2DsxyMAP, [485](#)
- mappel::Gauss2DsxyModel, [512](#)
- mappel::ImageFormat1DBase, [546](#)
- mappel::ImageFormat2DBase, [552](#)
- mappel::PoissonNoise1DObjective, [765](#)
- mappel::PoissonNoise2DObjective, [772](#)
- initial\_theta\_estimate
  - mappel::Gauss1DModel, [140](#)
  - mappel::Gauss1DModel, [168](#)
  - mappel::Gauss1DModel, [195](#), [196](#)
  - mappel::Gauss1DsMAP, [223](#)
  - mappel::Gauss1DsMLE, [251](#)
  - mappel::Gauss1DsModel, [279](#)
  - mappel::Gauss2DModel, [308](#)
  - mappel::Gauss2DModel, [338](#)
  - mappel::Gauss2DModel, [367](#), [368](#)
  - mappel::Gauss2DsMAP, [398](#), [399](#)
  - mappel::Gauss2DsMLE, [430](#), [431](#)
  - mappel::Gauss2DsModel, [462](#), [463](#)
  - mappel::Gauss2DsxyMAP, [491](#), [492](#)
  - mappel::Gauss2DsxyModel, [518](#)
- initialize\_arrays
  - mappel::estimator::ProfileBoundsData, [778](#)
  - mappel::estimator::ProfileBoundsDataStack, [781](#)
  - mappel::mcmc::MCMCData, [678](#)
  - mappel::mcmc::MCMCDataStack, [680](#)
  - mappel::mcmc::MCMCDebugData, [682](#)
- Install.md, [905](#)
- is\_negative\_definite
  - mappel, [36](#)
- is\_positive\_definite
  - mappel, [37](#)
- is\_symmetric
  - mappel, [37](#)
- istarts\_with
  - mappel, [37](#)
- IterativeMaximizer
  - mappel::estimator::IterativeMaximizer, [559](#)
- lambda\_term\_color
  - mappel, [38](#)
- last\_backtrack\_idx
  - mappel::estimator::IterativeMaximizer, [573](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [717](#)
  - mappel::estimator::NewtonMaximizer, [738](#)
  - mappel::estimator::QuasiNewtonMaximizer, [802](#)
  - mappel::estimator::TrustRegionMaximizer, [872](#)
- lbound
  - mappel::Gauss1DModel, [153](#)
  - mappel::Gauss1DModel, [181](#)
  - mappel::Gauss1DModel, [208](#)

- mappel::Gauss1DsMAP, 237
- mappel::Gauss1DsMLE, 265
- mappel::Gauss1DsModel, 293
- mappel::Gauss2DMAP, 322
- mappel::Gauss2DMLE, 352
- mappel::Gauss2DModel, 381
- mappel::Gauss2DsMAP, 413
- mappel::Gauss2DsMLE, 445
- mappel::Gauss2DsModel, 477
- mappel::Gauss2DsxyMAP, 504
- mappel::Gauss2DsxyModel, 527
- mappel::MAPEstimator, 591
- mappel::MCMCAdaptor1Ds, 635
- mappel::MCMCAdaptor1D, 617
- mappel::MCMCAdaptor2Ds, 672
- mappel::MCMCAdaptor2D, 653
- mappel::MLEstimator, 697
- mappel::PointEmitterModel, 759
- likelihood\_objective
  - mappel::methods, 63
- llh
  - mappel::methods::likelihood, 70
  - mappel::methods::objective, 76
- llh\_components
  - mappel::methods::likelihood::debug, 72
  - mappel::methods::objective::debug, 79
- llh\_stack
  - mappel::methods::objective::openmp, 82
- local\_maximize
  - mappel::estimator::IterativeMaximizer, 569
  - mappel::estimator::NewtonDiagonalMaximizer, 712
  - mappel::estimator::NewtonMaximizer, 733
  - mappel::estimator::QuasiNewtonMaximizer, 797
  - mappel::estimator::TrustRegionMaximizer, 867
- local\_profile\_maximize
  - mappel::estimator::IterativeMaximizer, 569
  - mappel::estimator::NewtonDiagonalMaximizer, 712
  - mappel::estimator::NewtonMaximizer, 733
  - mappel::estimator::QuasiNewtonMaximizer, 797
  - mappel::estimator::TrustRegionMaximizer, 867
- log\_likelihood
  - PoissonGaussianNoise2DObjective.h, 931
- log\_prior\_beta2\_const
  - mappel, 38
- log\_prior\_beta\_const
  - mappel, 38
- log\_prior\_gamma\_const
  - mappel, 38
- log\_prior\_normal\_const
  - mappel, 38
- log\_prior\_pareto\_const
  - mappel, 38
- LogicalError
  - mappel::LogicalError, 577
- MAPEstimator
  - mappel::MAPEstimator, 580
- MAPEstimator.h, 905
- MCMCAdaptor1D.cpp, 908
- MCMCAdaptor1D.h, 909
- MCMCAdaptor1Ds
  - mappel::MCMCAdaptor1Ds, 622, 623
- MCMCAdaptor1Ds.cpp, 909
- MCMCAdaptor1Ds.h, 910
- MCMCAdaptor1D
  - mappel::MCMCAdaptor1D, 604, 605
- MCMCAdaptor2D.cpp, 911
- MCMCAdaptor2D.h, 911
- MCMCAdaptor2Ds
  - mappel::MCMCAdaptor2Ds, 659
- MCMCAdaptor2Ds.cpp, 912
- MCMCAdaptor2Ds.h, 912
- MCMCAdaptor2D
  - mappel::MCMCAdaptor2D, 640, 641
- MCMCAdaptorBase
  - mappel::MCMCAdaptorBase, 675
- MCMCAdaptorBase.cpp, 913
- MCMCAdaptorBase.h, 913
- MLEstimator
  - mappel::MLEstimator, 686
- MLEstimator.h, 914
- make\_DX\_stencil
  - mappel, 38
- make\_DXS2\_stencil
  - mappel, 38
- make\_DXS\_stencil
  - mappel, 39
- make\_DXSX\_stencil
  - mappel, 39
- make\_G\_stencil
  - mappel, 39
- make\_X\_stencil
  - mappel, 39
- make\_d\_stencil
  - mappel, 38
- make\_default\_prior
  - mappel::Gauss1DMAP, 140
  - mappel::Gauss1DMLE, 168
  - mappel::Gauss1DModel, 196
  - mappel::Gauss1DsMAP, 223
  - mappel::Gauss1DsMLE, 251
  - mappel::Gauss1DsModel, 279
  - mappel::Gauss2DMAP, 309
  - mappel::Gauss2DMLE, 339
  - mappel::Gauss2DModel, 368
  - mappel::Gauss2DsMAP, 399
  - mappel::Gauss2DsMLE, 431
  - mappel::Gauss2DsModel, 463
  - mappel::Gauss2DsxyMAP, 492



- mappel::Gauss2DsxyModel, 518
- make\_default\_prior\_beta\_position
  - mappel::Gauss1DMAP, 141
  - mappel::Gauss1DMLE, 169
  - mappel::Gauss1DModel, 196
  - mappel::Gauss1DsMAP, 223
  - mappel::Gauss1DsMLE, 252
  - mappel::Gauss1DsModel, 280
  - mappel::Gauss2DMAP, 309
  - mappel::Gauss2DMLE, 339
  - mappel::Gauss2DModel, 368
  - mappel::Gauss2DsMAP, 399
  - mappel::Gauss2DsMLE, 431
  - mappel::Gauss2DsModel, 463
  - mappel::Gauss2DsxyMAP, 492
- make\_default\_prior\_normal\_position
  - mappel::Gauss1DMAP, 141
  - mappel::Gauss1DMLE, 169
  - mappel::Gauss1DModel, 196
  - mappel::Gauss1DsMAP, 224
  - mappel::Gauss1DsMLE, 252
  - mappel::Gauss1DsModel, 280
  - mappel::Gauss2DMAP, 309
  - mappel::Gauss2DMLE, 339
  - mappel::Gauss2DModel, 368
  - mappel::Gauss2DsMAP, 399
  - mappel::Gauss2DsMLE, 431
  - mappel::Gauss2DsModel, 463
  - mappel::Gauss2DsxyMAP, 492
- make\_estimator
  - mappel::methods, 63
  - PoissonGaussianNoise2DObjective.h, 931
- make\_gaussian\_stencil
  - mappel, 39
- make\_image
  - mappel::Gauss1DMAP, 141
  - mappel::Gauss1DMLE, 169
  - mappel::Gauss1DModel, 196
  - mappel::Gauss1DsMAP, 224
  - mappel::Gauss1DsMLE, 252
  - mappel::Gauss1DsModel, 280
  - mappel::Gauss2DMAP, 309
  - mappel::Gauss2DMLE, 339
  - mappel::Gauss2DModel, 368
  - mappel::Gauss2DsMAP, 399
  - mappel::Gauss2DsMLE, 432
  - mappel::Gauss2DsModel, 463
  - mappel::Gauss2DsxyMAP, 492
  - mappel::Gauss2DsxyModel, 518
  - mappel::ImageFormat1DBase, 547
  - mappel::ImageFormat2DBase, 554
  - mappel::PoissonNoise1DObjective, 768
  - mappel::PoissonNoise2DObjective, 775
- mappel::Gauss1DMAP, 141
- mappel::Gauss1DMLE, 169
- mappel::Gauss1DModel, 196
- mappel::Gauss1DsMAP, 224
- mappel::Gauss1DsMLE, 252
- mappel::Gauss1DsModel, 280
- mappel::Gauss2DMAP, 309
- mappel::Gauss2DMLE, 339
- mappel::Gauss2DModel, 369
- mappel::Gauss2DsMAP, 400
- mappel::Gauss2DsMLE, 432
- mappel::Gauss2DsModel, 464
- mappel::Gauss2DsxyMAP, 492
- mappel::Gauss2DsxyModel, 518
- mappel::ImageFormat1DBase, 547
- mappel::ImageFormat2DBase, 554
- mappel::PoissonNoise1DObjective, 768
- mappel::PoissonNoise2DObjective, 775
- make\_internal\_1Dsum\_estimator
  - mappel::Gauss2DMAP, 310
  - mappel::Gauss2DMLE, 340
  - mappel::Gauss2DModel, 369
  - mappel::Gauss2DsMAP, 400
  - mappel::Gauss2DsMLE, 432
  - mappel::Gauss2DsModel, 464
- make\_param
  - mappel::Gauss1DMAP, 141
  - mappel::Gauss1DMLE, 169
  - mappel::Gauss1DModel, 196, 197
  - mappel::Gauss1DsMAP, 224
  - mappel::Gauss1DsMLE, 252
  - mappel::Gauss1DsModel, 280
  - mappel::Gauss2DMAP, 310
  - mappel::Gauss2DMLE, 340
  - mappel::Gauss2DModel, 369
  - mappel::Gauss2DsMAP, 400
  - mappel::Gauss2DsMLE, 432
  - mappel::Gauss2DsModel, 464
  - mappel::Gauss2DsxyMAP, 493
  - mappel::Gauss2DsxyModel, 519
  - mappel::MAPEstimator, 584
  - mappel::MCMCAdaptor1Ds, 626, 627
  - mappel::MCMCAdaptor1D, 609
  - mappel::MCMCAdaptor2Ds, 663
  - mappel::MCMCAdaptor2D, 644, 645
  - mappel::MLEstimator, 690
  - mappel::PointEmitterModel, 753
- make\_param\_mat
  - mappel::Gauss1DMAP, 142
  - mappel::Gauss1DMLE, 170
  - mappel::Gauss1DModel, 197
  - mappel::Gauss1DsMAP, 224, 225
  - mappel::Gauss1DsMLE, 253
  - mappel::Gauss1DsModel, 281
- make\_image\_stack

- mappel::Gauss2DModel, 369
- mappel::Gauss2DsMAP, 400
- mappel::Gauss2DsMLE, 433
- mappel::Gauss2DsModel, 464
- mappel::Gauss2DsxyMAP, 493
- mappel::Gauss2DsxyModel, 519
- mappel::MAPEstimator, 584
- mappel::MCMCAdaptor1Ds, 627
- mappel::MCMCAdaptor1D, 609
- mappel::MCMCAdaptor2Ds, 663
- mappel::MCMCAdaptor2D, 645
- mappel::MLEstimator, 690
- mappel::PointEmitterModel, 753
- make\_param\_mat\_stack
  - mappel::Gauss1DModel, 197
  - mappel::Gauss1DsMAP, 225
  - mappel::Gauss1DsMLE, 253
  - mappel::Gauss1DsModel, 281
  - mappel::Gauss2DModel, 370
  - mappel::Gauss2DsMAP, 401
  - mappel::Gauss2DsMLE, 433
  - mappel::Gauss2DsModel, 465
  - mappel::Gauss2DsxyMAP, 493
  - mappel::Gauss2DsxyModel, 519
  - mappel::MAPEstimator, 584
  - mappel::MCMCAdaptor1Ds, 627
  - mappel::MCMCAdaptor1D, 609
  - mappel::MCMCAdaptor2Ds, 663, 664
  - mappel::MCMCAdaptor2D, 645
  - mappel::MLEstimator, 690
  - mappel::PointEmitterModel, 753
- make\_param\_stack
  - mappel::Gauss1DModel, 197
  - mappel::Gauss1DsMAP, 225
  - mappel::Gauss1DsMLE, 253
  - mappel::Gauss1DsModel, 281
  - mappel::Gauss2DModel, 370
  - mappel::Gauss2DsMAP, 401
  - mappel::Gauss2DsMLE, 433
  - mappel::Gauss2DsModel, 465
  - mappel::Gauss2DsxyMAP, 493, 494
  - mappel::Gauss2DsxyModel, 519, 520
  - mappel::MAPEstimator, 584, 585
  - mappel::MCMCAdaptor1Ds, 627
- mappel::MCMCAdaptor1D, 609, 610
- mappel::MCMCAdaptor2Ds, 664
- mappel::MCMCAdaptor2D, 645
- mappel::MLEstimator, 690, 691
- mappel::PointEmitterModel, 753, 754
- make\_prior\_beta\_position
  - mappel::Gauss1DModel, 198
  - mappel::Gauss1DsMAP, 225
  - mappel::Gauss1DsMLE, 253
  - mappel::Gauss1DsModel, 281
  - mappel::Gauss2DModel, 370
  - mappel::Gauss2DsMAP, 401
  - mappel::Gauss2DsMLE, 433
  - mappel::Gauss2DsModel, 465
  - mappel::Gauss2DsxyMAP, 494
  - mappel::Gauss2DsxyModel, 520
- make\_prior\_component\_intensity
  - mappel::Gauss1DModel, 198
  - mappel::Gauss1DsMAP, 225
  - mappel::Gauss1DsMLE, 254
  - mappel::Gauss1DsModel, 282
  - mappel::Gauss2DModel, 370
  - mappel::Gauss2DsMAP, 401
  - mappel::Gauss2DsMLE, 434
  - mappel::Gauss2DsModel, 465
  - mappel::Gauss2DsxyMAP, 494
  - mappel::Gauss2DsxyModel, 520
  - mappel::MAPEstimator, 585
  - mappel::MCMCAdaptor1Ds, 628
  - mappel::MCMCAdaptor1D, 610
  - mappel::MCMCAdaptor2Ds, 664
  - mappel::MCMCAdaptor2D, 646
  - mappel::MLEstimator, 691
  - mappel::PointEmitterModel, 754
- make\_prior\_component\_position\_beta
  - mappel::Gauss1DModel, 198
  - mappel::Gauss1DsMAP, 226
  - mappel::Gauss1DsMLE, 254
  - mappel::Gauss1DsModel, 282
  - mappel::Gauss2DModel, 371
  - mappel::Gauss2DsMAP, 402
  - mappel::Gauss2DsMLE, 434



- mappel::Gauss2DsModel, [466](#)
- mappel::Gauss2DsxyMAP, [494](#)
- mappel::Gauss2DsxyModel, [520](#)
- mappel::MAPEstimator, [585](#)
- mappel::MCMCAdaptor1Ds, [628](#)
- mappel::MCMCAdaptor1D, [610](#)
- mappel::MCMCAdaptor2Ds, [664](#)
- mappel::MCMCAdaptor2D, [646](#)
- mappel::MLEstimator, [691](#)
- mappel::PointEmitterModel, [754](#)
- make\_prior\_component\_position\_normal
  - mappel::Gauss1DMAP, [143](#)
  - mappel::Gauss1DMLE, [171](#)
  - mappel::Gauss1DModel, [198](#)
  - mappel::Gauss1DsMAP, [226](#)
  - mappel::Gauss1DsMLE, [254](#)
  - mappel::Gauss1DsModel, [282](#)
  - mappel::Gauss2DMAP, [312](#)
  - mappel::Gauss2DMLE, [342](#)
  - mappel::Gauss2DModel, [371](#)
  - mappel::Gauss2DsMAP, [402](#)
  - mappel::Gauss2DsMLE, [434](#)
  - mappel::Gauss2DsModel, [466](#)
  - mappel::Gauss2DsxyMAP, [494](#)
  - mappel::Gauss2DsxyModel, [520](#)
  - mappel::MAPEstimator, [585](#)
  - mappel::MCMCAdaptor1Ds, [628](#)
  - mappel::MCMCAdaptor1D, [610](#)
  - mappel::MCMCAdaptor2Ds, [665](#)
  - mappel::MCMCAdaptor2D, [646](#)
  - mappel::MLEstimator, [691](#)
  - mappel::PointEmitterModel, [754](#)
- make\_prior\_component\_sigma
  - mappel::Gauss1DMAP, [143](#)
  - mappel::Gauss1DMLE, [171](#)
  - mappel::Gauss1DModel, [198](#)
  - mappel::Gauss1DsMAP, [226](#)
  - mappel::Gauss1DsMLE, [254](#)
  - mappel::Gauss1DsModel, [282](#)
  - mappel::Gauss2DMAP, [312](#)
  - mappel::Gauss2DMLE, [342](#)
  - mappel::Gauss2DModel, [371](#)
  - mappel::Gauss2DsMAP, [402](#)
  - mappel::Gauss2DsMLE, [434](#)
  - mappel::Gauss2DsModel, [466](#)
  - mappel::Gauss2DsxyMAP, [495](#)
  - mappel::Gauss2DsxyModel, [521](#)
  - mappel::MAPEstimator, [585](#)
  - mappel::MCMCAdaptor1Ds, [628](#)
  - mappel::MCMCAdaptor1D, [610](#)
  - mappel::MCMCAdaptor2Ds, [665](#)
  - mappel::MCMCAdaptor2D, [646](#)
  - mappel::MLEstimator, [691](#)
  - mappel::PointEmitterModel, [754](#)
- make\_prior\_normal\_position
  - mappel::Gauss1DMAP, [144](#)
  - mappel::Gauss1DMLE, [172](#)
  - mappel::Gauss1DModel, [199](#)
  - mappel::Gauss1DsMAP, [226](#)
  - mappel::Gauss1DsMLE, [255](#)
  - mappel::Gauss1DsModel, [283](#)
  - mappel::Gauss2DMAP, [312](#)
  - mappel::Gauss2DMLE, [342](#)
  - mappel::Gauss2DModel, [371](#)
  - mappel::Gauss2DsMAP, [402](#)
  - mappel::Gauss2DsMLE, [435](#)
  - mappel::Gauss2DsModel, [466](#)
  - mappel::Gauss2DsxyMAP, [495](#)
  - mappel::Gauss2DsxyModel, [521](#)
- make\_stencil
  - mappel::Gauss1DMAP, [144](#)
  - mappel::Gauss1DMLE, [172](#)
  - mappel::Gauss1DModel, [199](#)
  - mappel::Gauss1DsMAP, [227](#)
  - mappel::Gauss1DsMLE, [255](#)
  - mappel::Gauss1DsModel, [283](#)
  - mappel::Gauss2DMAP, [312](#)
  - mappel::Gauss2DMLE, [342](#)
  - mappel::Gauss2DModel, [371](#)
  - mappel::Gauss2DsMAP, [403](#)
  - mappel::Gauss2DsMLE, [435](#)
  - mappel::Gauss2DsModel, [466](#)
  - mappel::Gauss2DsxyMAP, [495](#)
  - mappel::Gauss2DsxyModel, [521](#)
- make\_unique
  - mappel, [39](#)
- mappel, [20](#)
  - beta2\_prior\_grad, [29](#)
  - beta2\_prior\_grad2, [29](#)
  - beta\_prior\_grad, [29](#)
  - beta\_prior\_grad2, [29](#)
  - BoolVecT, [26](#)
  - BoolT, [26](#)
  - cgauss\_compute\_estimate, [29](#), [30](#)
  - cgauss\_compute\_estimate\_debug, [30](#)
  - cgauss\_heuristic\_compute\_estimate, [30](#), [31](#)
  - check\_lower\_bound\_hyperparameter, [31](#)
  - check\_positive\_hyperparameter, [31](#)
  - check\_unit\_hyperparameter, [31](#)
  - chisq\_quantile, [31](#)
  - cholesky, [31](#)
  - cholesky\_convert\_full\_matrix, [32](#)
  - cholesky\_convert\_lower\_triangular, [32](#)
  - cholesky\_make\_negative\_definite, [32](#)
  - cholesky\_make\_positive\_definite, [32](#)
  - cholesky\_solve, [32](#)
  - clamp, [33](#)
  - copy\_Lsym\_mat, [33](#)

copy\_Usym\_mat, 33  
copy\_Usym\_mat\_stack, 33  
CubeT, 27  
enable\_all\_cpus, 33  
EnableIfSubclassT, 27  
estimate\_background, 33, 34  
estimate\_gaussian\_2Dmax, 34  
estimate\_gaussian\_3Dmax, 34  
estimate\_intensity, 34  
fill\_DX\_stencil, 34  
fill\_DXS2\_stencil, 34  
fill\_DXS\_stencil, 34  
fill\_DXSX\_stencil, 35  
fill\_G\_stencil, 35  
fill\_X\_stencil, 35  
fill\_d\_stencil, 34  
fill\_gaussian\_stencil, 35  
gamma\_prior\_grad, 35  
gamma\_prior\_grad2, 35  
gauss\_norm, 35  
gaussian\_3D\_convolution, 35  
gaussian\_convolution, 36  
generate\_poisson, 36  
generate\_poisson\_large, 36  
generate\_poisson\_small, 36  
icontains, 36  
IdxMatT, 27  
IdxVecT, 27  
IdxT, 27  
ImageCoordT, 27  
ImagePixelT, 27  
ImageStackT, 27  
ImageT, 27  
is\_negative\_definite, 36  
is\_positive\_definite, 37  
is\_symmetric, 37  
istarts\_with, 37  
lambda\_term\_color, 38  
log\_prior\_beta2\_const, 38  
log\_prior\_beta\_const, 38  
log\_prior\_gamma\_const, 38  
log\_prior\_normal\_const, 38  
log\_prior\_pareto\_const, 38  
make\_DX\_stencil, 38  
make\_DXS2\_stencil, 38  
make\_DXS\_stencil, 39  
make\_DXSX\_stencil, 39  
make\_G\_stencil, 39  
make\_X\_stencil, 39  
make\_d\_stencil, 38  
make\_gaussian\_stencil, 39  
make\_unique, 39  
MappelError, 27  
MatT, 28  
maxidx, 40  
ModelDataStackT, 28  
ModelDataT, 28  
modified\_cholesky, 40  
norm\_sq, 40  
normal\_prior\_grad, 40  
normal\_prior\_grad2, 40  
normal\_quantile\_onesided, 40  
normal\_quantile\_twosided, 41  
operator<<, 41  
ParallelRngGeneratorT, 28  
ParallelRngManagerT, 28  
ParamVecT, 28  
ParamT, 28  
pareto\_prior\_grad, 42  
pareto\_prior\_grad2, 42  
poisson\_log\_likelihood, 42  
print\_centered\_title, 42  
print\_image, 42  
print\_labeled\_image, 43  
print\_text\_image, 43  
print\_vec\_row, 43  
refine\_gaussian\_2Dmax, 43  
refine\_gaussian\_3Dmax, 43  
relative\_poisson\_log\_likelihood, 44  
restrict\_value\_range, 44  
ReturnIfSubclassT, 28  
rllh\_beta2\_prior, 44  
rllh\_beta\_prior, 44  
rllh\_gamma\_prior, 44  
rllh\_normal\_prior, 44  
rllh\_pareto\_prior, 44  
rng\_manager, 45  
RngSeedT, 28  
sgn, 44  
square, 44  
StatsT, 28  
StencilVecT, 29  
StencilT, 28  
StringVecT, 29  
TERM\_BLACK, 45  
TERM\_BLUE, 45  
TERM\_CYAN, 45  
TERM\_DIM\_BLACK, 45  
TERM\_DIM\_BLUE, 45  
TERM\_DIM\_CYAN, 45  
TERM\_DIM\_GREEN, 46  
TERM\_DIM\_MAGENTA, 46  
TERM\_DIM\_RED, 46  
TERM\_DIM\_WHITE, 46  
TERM\_DIM\_YELLOW, 46  
TERM\_GREEN, 46  
TERM\_MAGENTA, 46  
TERM\_RED, 46

- TERM\_WHITE, [47](#)
- TERM\_YELLOW, [47](#)
- UniformDistT, [29](#)
- VecFieldT, [29](#)
- VecT, [29](#)
- mappel::ArrayShapeError, [91](#)
  - ArrayShapeError, [92](#)
- mappel::ArraySizeError, [92](#)
  - ArraySizeError, [92](#)
- mappel::Gauss1DMAP, [128](#)
  - bound\_theta, [135](#)
  - bounded\_theta, [135](#)
  - bounded\_theta\_stack, [135](#)
  - bounds\_epsilon, [150](#)
  - check\_image\_shape, [136](#)
  - check\_param\_shape, [136](#)
  - check\_psf\_sigma, [136](#)
  - check\_size, [137](#)
  - default\_alpha\_sigma, [150](#)
  - default\_beta\_pos, [150](#)
  - default\_intensity\_kappa, [151](#)
  - default\_max\_l, [151](#)
  - default\_mean\_l, [151](#)
  - default\_pixel\_mean\_bg, [151](#)
  - default\_sigma\_pos, [151](#)
  - DefaultPriorType, [151](#)
  - DefaultSeperableInitEstimator, [152](#)
  - estimator\_names, [152](#)
  - eta\_bg, [152](#)
  - eta\_l, [152](#)
  - eta\_x, [152](#)
  - Gauss1DMAP, [135](#)
  - get\_hyperparam\_index, [137](#)
  - get\_hyperparam\_names, [137](#)
  - get\_hyperparam\_value, [137](#)
  - get\_hyperparams, [137](#)
  - get\_image\_from\_stack, [137](#)
  - get\_lbound, [137](#)
  - get\_mcmc\_num\_phases, [138](#)
  - get\_mcmc\_sigma\_scale, [138](#)
  - get\_num\_hyperparams, [138](#)
  - get\_num\_params, [138](#)
  - get\_num\_pixels, [138](#)
  - get\_param\_names, [138](#)
  - get\_prior, [138](#), [139](#)
  - get\_psf\_sigma, [139](#)
  - get\_rng\_generator, [139](#)
  - get\_rng\_manager, [139](#)
  - get\_size, [139](#)
  - get\_size\_image\_stack, [140](#)
  - get\_stats, [140](#)
  - get\_ubound, [140](#)
  - global\_default\_mcmc\_sigma\_scale, [152](#)
  - global\_max\_mcmc\_sigma\_scale, [153](#)
  - global\_max\_psf\_sigma, [153](#)
  - global\_max\_size, [153](#)
  - global\_min\_psf\_sigma, [153](#)
  - global\_min\_size, [153](#)
  - has\_hyperparam, [140](#)
  - ImageCoordT, [133](#)
  - ImagePixelT, [133](#)
  - ImageShapeT, [133](#)
  - ImageSizeShapeT, [133](#)
  - ImageSizeVecShapeT, [133](#)
  - ImageSizeVecT, [133](#)
  - ImageSizeT, [133](#)
  - ImageStackShapeT, [133](#)
  - ImageStackT, [134](#)
  - ImageT, [134](#)
  - initial\_theta\_estimate, [140](#)
  - lbound, [153](#)
  - make\_default\_prior, [140](#)
  - make\_default\_prior\_beta\_position, [141](#)
  - make\_default\_prior\_normal\_position, [141](#)
  - make\_image, [141](#)
  - make\_image\_stack, [141](#)
  - make\_param, [141](#)
  - make\_param\_mat, [142](#)
  - make\_param\_mat\_stack, [142](#)
  - make\_param\_stack, [142](#)
  - make\_prior\_beta\_position, [142](#)
  - make\_prior\_component\_intensity, [143](#)
  - make\_prior\_component\_position\_beta, [143](#)
  - make\_prior\_component\_position\_normal, [143](#)
  - make\_prior\_component\_sigma, [143](#)
  - make\_prior\_normal\_position, [144](#)
  - make\_stencil, [144](#)
  - ModelDataStackT, [134](#)
  - ModelDataT, [134](#)
  - name, [154](#)
  - num\_dim, [154](#)
  - num\_hyperparams, [154](#)
  - num\_params, [154](#)
  - num\_phases, [154](#)
  - operator=, [144](#), [145](#)
  - ParamVecT, [134](#)
  - ParamT, [134](#)
  - pixel\_grad, [145](#)
  - pixel\_grad2, [145](#)
  - pixel\_hess, [145](#)
  - pixel\_hess\_update, [145](#)
  - pixel\_model\_value, [145](#)
  - prior, [154](#)
  - prior\_types, [155](#)
  - psf\_sigma, [155](#)
  - reflected\_theta, [146](#)
  - reflected\_theta\_stack, [146](#)
  - rename\_hyperparam, [146](#)

- sample\_mcmc\_candidate, 146
- sample\_prior, 146, 147
- set\_background\_mcmc\_sampling, 147
- set\_bounds, 147
- set\_hyperparam\_names, 147
- set\_hyperparam\_value, 147
- set\_hyperparams, 147
- set\_image\_in\_stack, 148
- set\_intensity\_mcmc\_sampling, 148
- set\_lbound, 148
- set\_mcmc\_num\_phases, 148
- set\_mcmc\_sigma\_scale, 148
- set\_param\_names, 148
- set\_prior, 149
- set\_psf\_sigma, 149
- set\_rng\_seed, 149
- set\_size, 149
- set\_ubound, 150
- sigma\_scale, 155
- size, 155
- StencilVecT, 134
- theta\_in\_bounds, 150
- theta\_stack\_in\_bounds, 150
- ubound, 156
- mappel::Gauss1DMLE, 156
  - bound\_theta, 163
  - bounded\_theta, 163
  - bounded\_theta\_stack, 163
  - bounds\_epsilon, 178
  - check\_image\_shape, 164
  - check\_param\_shape, 164
  - check\_psf\_sigma, 164
  - check\_size, 165
  - default\_alpha\_sigma, 178
  - default\_beta\_pos, 178
  - default\_intensity\_kappa, 179
  - default\_max\_l, 179
  - default\_mean\_l, 179
  - default\_pixel\_mean\_bg, 179
  - default\_sigma\_pos, 179
  - DefaultPriorType, 179
  - DefaultSeperableInitEstimator, 180
  - estimator\_names, 180
  - eta\_bg, 180
  - eta\_l, 180
  - eta\_x, 180
  - Gauss1DMLE, 163
  - get\_hyperparam\_index, 165
  - get\_hyperparam\_names, 165
  - get\_hyperparam\_value, 165
  - get\_hyperparams, 165
  - get\_image\_from\_stack, 165
  - get\_lbound, 165
  - get\_mcmc\_num\_phases, 166
  - get\_mcmc\_sigma\_scale, 166
  - get\_num\_hyperparams, 166
  - get\_num\_params, 166
  - get\_num\_pixels, 166
  - get\_param\_names, 166
  - get\_prior, 166, 167
  - get\_psf\_sigma, 167
  - get\_rng\_generator, 167
  - get\_rng\_manager, 167
  - get\_size, 167
  - get\_size\_image\_stack, 168
  - get\_stats, 168
  - get\_ubound, 168
  - global\_default\_mcmc\_sigma\_scale, 180
  - global\_max\_mcmc\_sigma\_scale, 181
  - global\_max\_psf\_sigma, 181
  - global\_max\_size, 181
  - global\_min\_psf\_sigma, 181
  - global\_min\_size, 181
  - has\_hyperparam, 168
  - ImageCoordT, 161
  - ImagePixelT, 161
  - ImageShapeT, 161
  - ImageSizeShapeT, 161
  - ImageSizeVecShapeT, 161
  - ImageSizeVecT, 161
  - ImageSizeT, 161
  - ImageStackShapeT, 161
  - ImageStackT, 162
  - ImageT, 162
  - initial\_theta\_estimate, 168
  - lbound, 181
  - make\_default\_prior, 168
  - make\_default\_prior\_beta\_position, 169
  - make\_default\_prior\_normal\_position, 169
  - make\_image, 169
  - make\_image\_stack, 169
  - make\_param, 169
  - make\_param\_mat, 170
  - make\_param\_mat\_stack, 170
  - make\_param\_stack, 170
  - make\_prior\_beta\_position, 170
  - make\_prior\_component\_intensity, 171
  - make\_prior\_component\_position\_beta, 171
  - make\_prior\_component\_position\_normal, 171
  - make\_prior\_component\_sigma, 171
  - make\_prior\_normal\_position, 172
  - make\_stencil, 172
  - ModelDataStackT, 162
  - ModelDataT, 162
  - name, 182
  - num\_dim, 182
  - num\_hyperparams, 182
  - num\_params, 182

- num\_phases, [182](#)
- operator=, [172](#), [173](#)
- ParamVecT, [162](#)
- ParamT, [162](#)
- pixel\_grad, [173](#)
- pixel\_grad2, [173](#)
- pixel\_hess, [173](#)
- pixel\_hess\_update, [173](#)
- pixel\_model\_value, [173](#)
- prior, [182](#)
- prior\_types, [183](#)
- psf\_sigma, [183](#)
- reflected\_theta, [174](#)
- reflected\_theta\_stack, [174](#)
- rename\_hyperparam, [174](#)
- sample\_mcmc\_candidate, [174](#)
- sample\_prior, [174](#), [175](#)
- set\_background\_mcmc\_sampling, [175](#)
- set\_bounds, [175](#)
- set\_hyperparam\_names, [175](#)
- set\_hyperparam\_value, [175](#)
- set\_hyperparams, [175](#)
- set\_image\_in\_stack, [176](#)
- set\_intensity\_mcmc\_sampling, [176](#)
- set\_lbound, [176](#)
- set\_mcmc\_num\_phases, [176](#)
- set\_mcmc\_sigma\_scale, [176](#)
- set\_param\_names, [176](#)
- set\_prior, [177](#)
- set\_psf\_sigma, [177](#)
- set\_rng\_seed, [177](#)
- set\_size, [177](#)
- set\_ubound, [178](#)
- sigma\_scale, [183](#)
- size, [183](#)
- StencilVecT, [162](#)
- theta\_in\_bounds, [178](#)
- theta\_stack\_in\_bounds, [178](#)
- ubound, [184](#)
- mappel::Gauss1DModel, [184](#)
  - bound\_theta, [190](#)
  - bounded\_theta, [190](#)
  - bounded\_theta\_stack, [191](#)
  - bounds\_epsilon, [205](#)
  - check\_image\_shape, [191](#)
  - check\_param\_shape, [191](#)
  - check\_psf\_sigma, [191](#), [192](#)
  - check\_size, [192](#)
  - default\_alpha\_sigma, [205](#)
  - default\_beta\_pos, [205](#)
  - default\_intensity\_kappa, [205](#)
  - default\_max\_l, [205](#)
  - default\_mean\_l, [206](#)
  - default\_pixel\_mean\_bg, [206](#)
  - default\_sigma\_pos, [206](#)
  - DefaultPriorType, [206](#)
  - DefaultSeperableInitEstimator, [206](#)
  - eta\_bg, [206](#)
  - eta\_l, [207](#)
  - eta\_x, [207](#)
  - Gauss1DModel, [190](#)
  - get\_hyperparam\_index, [192](#)
  - get\_hyperparam\_names, [192](#)
  - get\_hyperparam\_value, [192](#)
  - get\_hyperparams, [192](#)
  - get\_image\_from\_stack, [193](#)
  - get\_lbound, [193](#)
  - get\_mcmc\_num\_phases, [193](#)
  - get\_mcmc\_sigma\_scale, [193](#)
  - get\_num\_hyperparams, [193](#)
  - get\_num\_params, [193](#)
  - get\_num\_pixels, [193](#)
  - get\_param\_names, [194](#)
  - get\_prior, [194](#)
  - get\_psf\_sigma, [194](#)
  - get\_rng\_generator, [194](#)
  - get\_rng\_manager, [194](#)
  - get\_size, [195](#)
  - get\_size\_image\_stack, [195](#)
  - get\_stats, [195](#)
  - get\_ubound, [195](#)
  - global\_default\_mcmc\_sigma\_scale, [207](#)
  - global\_max\_mcmc\_sigma\_scale, [207](#)
  - global\_max\_psf\_sigma, [207](#)
  - global\_max\_size, [208](#)
  - global\_min\_psf\_sigma, [208](#)
  - global\_min\_size, [208](#)
  - has\_hyperparam, [195](#)
  - ImageCoordT, [188](#)
  - ImagePixelT, [188](#)
  - ImageShapeT, [188](#)
  - ImageSizeShapeT, [189](#)
  - ImageSizeVecShapeT, [189](#)
  - ImageSizeVecT, [189](#)
  - ImageSizeT, [189](#)
  - ImageStackShapeT, [189](#)
  - ImageStackT, [189](#)
  - ImageT, [189](#)
  - initial\_theta\_estimate, [195](#), [196](#)
  - lbound, [208](#)
  - make\_default\_prior, [196](#)
  - make\_default\_prior\_beta\_position, [196](#)
  - make\_default\_prior\_normal\_position, [196](#)
  - make\_image, [196](#)
  - make\_image\_stack, [196](#)
  - make\_param, [196](#), [197](#)
  - make\_param\_mat, [197](#)
  - make\_param\_mat\_stack, [197](#)

- make\_param\_stack, 197
- make\_prior\_beta\_position, 198
- make\_prior\_component\_intensity, 198
- make\_prior\_component\_position\_beta, 198
- make\_prior\_component\_position\_normal, 198
- make\_prior\_component\_sigma, 198
- make\_prior\_normal\_position, 199
- make\_stencil, 199
- num\_dim, 208
- num\_hyperparams, 209
- num\_params, 209
- num\_phases, 209
- operator=, 199, 200
- ParamVecT, 190
- ParamT, 190
- pixel\_grad, 200
- pixel\_grad2, 200
- pixel\_hess, 200
- pixel\_hess\_update, 200
- pixel\_model\_value, 200
- prior, 209
- prior\_types, 209
- psf\_sigma, 210
- reflected\_theta, 200
- reflected\_theta\_stack, 201
- rename\_hyperparam, 201
- sample\_mcmc\_candidate, 201
- sample\_prior, 201
- set\_background\_mcmc\_sampling, 202
- set\_bounds, 202
- set\_hyperparam\_names, 202
- set\_hyperparam\_value, 202
- set\_hyperparams, 202
- set\_image\_in\_stack, 202
- set\_intensity\_mcmc\_sampling, 203
- set\_lbound, 203
- set\_mcmc\_num\_phases, 203
- set\_mcmc\_sigma\_scale, 203
- set\_param\_names, 203
- set\_prior, 203
- set\_psf\_sigma, 204
- set\_rng\_seed, 204
- set\_size, 204
- set\_ubound, 204
- sigma\_scale, 210
- size, 210
- StencilVecT, 190
- theta\_in\_bounds, 204
- theta\_stack\_in\_bounds, 205
- ubound, 210
- mappel::Gauss1DModel::Stencil, 836
  - bg, 838
  - compute\_derivatives, 838
  - DXS, 839
  - derivatives\_computed, 839
  - DX, 839
  - dx, 839
  - Gx, 839
  - I, 838
  - model, 840
  - operator<<, 839
  - ParamT, 837
  - Stencil, 838
  - theta, 840
  - X, 840
  - x, 838
- mappel::Gauss1DsMAP, 211
  - bound\_theta, 218
  - bounded\_theta, 218
  - bounded\_theta\_stack, 218
  - bounds\_epsilon, 233
  - check\_image\_shape, 218
  - check\_param\_shape, 219
  - check\_psf\_sigma, 219
  - check\_size, 219
  - default\_alpha\_sigma, 233
  - default\_beta\_pos, 234
  - default\_intensity\_kappa, 234
  - default\_max\_I, 234
  - default\_mean\_I, 234
  - default\_pixel\_mean\_bg, 234
  - default\_sigma\_pos, 234
  - DefaultPriorType, 235
  - DefaultSeperableInitEstimator, 235
  - estimator\_names, 235
  - eta\_bg, 235
  - eta\_I, 235
  - eta\_sigma, 235
  - eta\_x, 236
  - Gauss1DsMAP, 217, 218
  - get\_hyperparam\_index, 219
  - get\_hyperparam\_names, 220
  - get\_hyperparam\_value, 220
  - get\_hyperparams, 220
  - get\_image\_from\_stack, 220
  - get\_lbound, 220
  - get\_max\_sigma, 220
  - get\_mcmc\_num\_phases, 220
  - get\_mcmc\_sigma\_scale, 221
  - get\_min\_sigma, 221
  - get\_num\_hyperparams, 221
  - get\_num\_params, 221
  - get\_num\_pixels, 221
  - get\_param\_names, 221
  - get\_prior, 221, 222
  - get\_rng\_generator, 222
  - get\_rng\_manager, 222
  - get\_size, 222

get\_size\_image\_stack, 222  
get\_stats, 222  
get\_ubound, 223  
global\_default\_mcmc\_sigma\_scale, 236  
global\_max\_mcmc\_sigma\_scale, 236  
global\_max\_psf\_sigma, 236  
global\_max\_size, 236  
global\_min\_psf\_sigma, 236  
global\_min\_size, 237  
has\_hyperparam, 223  
ImageCoordT, 215  
ImagePixelT, 215  
ImageShapeT, 215  
ImageSizeShapeT, 215  
ImageSizeVecShapeT, 216  
ImageSizeVecT, 216  
ImageSizeT, 216  
ImageStackShapeT, 216  
ImageStackT, 216  
ImageT, 216  
initial\_theta\_estimate, 223  
lbound, 237  
make\_default\_prior, 223  
make\_default\_prior\_beta\_position, 223  
make\_default\_prior\_normal\_position, 224  
make\_image, 224  
make\_image\_stack, 224  
make\_param, 224  
make\_param\_mat, 224, 225  
make\_param\_mat\_stack, 225  
make\_param\_stack, 225  
make\_prior\_beta\_position, 225  
make\_prior\_component\_intensity, 225  
make\_prior\_component\_position\_beta, 226  
make\_prior\_component\_position\_normal, 226  
make\_prior\_component\_sigma, 226  
make\_prior\_normal\_position, 226  
make\_stencil, 227  
ModelDataStackT, 216  
ModelDataT, 217  
name, 237  
num\_dim, 237  
num\_hyperparams, 237  
num\_params, 237  
num\_phases, 238  
operator=, 227  
ParamVecT, 217  
ParamT, 217  
pixel\_grad, 228  
pixel\_grad2, 228  
pixel\_hess, 228  
pixel\_hess\_update, 228  
pixel\_model\_value, 228  
prior, 238

prior\_types, 238  
reflected\_theta, 228  
reflected\_theta\_stack, 229  
rename\_hyperparam, 229  
sample\_mcmc\_candidate, 229  
sample\_prior, 229  
set\_background\_mcmc\_sampling, 230  
set\_bounds, 230  
set\_hyperparam\_names, 230  
set\_hyperparam\_value, 230  
set\_hyperparams, 230  
set\_image\_in\_stack, 230  
set\_intensity\_mcmc\_sampling, 231  
set\_lbound, 231  
set\_max\_sigma, 231  
set\_mcmc\_num\_phases, 231  
set\_mcmc\_sigma\_scale, 231  
set\_min\_sigma, 232  
set\_param\_names, 232  
set\_prior, 232  
set\_rng\_seed, 232  
set\_size, 232, 233  
set\_ubound, 233  
sigma\_scale, 238  
size, 239  
StencilVecT, 217  
theta\_in\_bounds, 233  
theta\_stack\_in\_bounds, 233  
ubound, 239  
mappel::Gauss1DsMLE, 239  
bound\_theta, 246  
bounded\_theta, 246  
bounded\_theta\_stack, 246  
bounds\_epsilon, 262  
check\_image\_shape, 247  
check\_param\_shape, 247  
check\_psf\_sigma, 247  
check\_size, 248  
default\_alpha\_sigma, 262  
default\_beta\_pos, 262  
default\_intensity\_kappa, 262  
default\_max\_l, 262  
default\_mean\_l, 262  
default\_pixel\_mean\_bg, 262  
default\_sigma\_pos, 263  
DefaultPriorType, 263  
DefaultSeperableInitEstimator, 263  
estimator\_names, 263  
eta\_bg, 263  
eta\_l, 263  
eta\_sigma, 264  
eta\_x, 264  
Gauss1DsMLE, 246  
get\_hyperparam\_index, 248



get\_hyperparam\_names, 248  
get\_hyperparam\_value, 248  
get\_hyperparams, 248  
get\_image\_from\_stack, 248  
get\_lbound, 248  
get\_max\_sigma, 249  
get\_mcmc\_num\_phases, 249  
get\_mcmc\_sigma\_scale, 249  
get\_min\_sigma, 249  
get\_num\_hyperparams, 249  
get\_num\_params, 249  
get\_num\_pixels, 249  
get\_param\_names, 250  
get\_prior, 250  
get\_rng\_generator, 250  
get\_rng\_manager, 250  
get\_size, 250  
get\_size\_image\_stack, 251  
get\_stats, 251  
get\_ubound, 251  
global\_default\_mcmc\_sigma\_scale, 264  
global\_max\_mcmc\_sigma\_scale, 264  
global\_max\_psf\_sigma, 264  
global\_max\_size, 264  
global\_min\_psf\_sigma, 265  
global\_min\_size, 265  
has\_hyperparam, 251  
ImageCoordT, 244  
ImagePixelT, 244  
ImageShapeT, 244  
ImageSizeShapeT, 244  
ImageSizeVecShapeT, 244  
ImageSizeVecT, 244  
ImageSizeT, 244  
ImageStackShapeT, 244  
ImageStackT, 245  
ImageT, 245  
initial\_theta\_estimate, 251  
lbound, 265  
make\_default\_prior, 251  
make\_default\_prior\_beta\_position, 252  
make\_default\_prior\_normal\_position, 252  
make\_image, 252  
make\_image\_stack, 252  
make\_param, 252  
make\_param\_mat, 253  
make\_param\_mat\_stack, 253  
make\_param\_stack, 253  
make\_prior\_beta\_position, 253  
make\_prior\_component\_intensity, 254  
make\_prior\_component\_position\_beta, 254  
make\_prior\_component\_position\_normal, 254  
make\_prior\_component\_sigma, 254  
make\_prior\_normal\_position, 255  
make\_stencil, 255  
ModelDataStackT, 245  
ModelDataT, 245  
name, 265  
num\_dim, 265  
num\_hyperparams, 265  
num\_params, 266  
num\_phases, 266  
operator=, 255, 256  
ParamVecT, 245  
ParamT, 245  
pixel\_grad, 256  
pixel\_grad2, 256  
pixel\_hess, 256  
pixel\_hess\_update, 256  
pixel\_model\_value, 256  
prior, 266  
prior\_types, 266  
reflected\_theta, 257  
reflected\_theta\_stack, 257  
rename\_hyperparam, 257  
sample\_mcmc\_candidate, 257  
sample\_prior, 257, 258  
set\_background\_mcmc\_sampling, 258  
set\_bounds, 258  
set\_hyperparam\_names, 258  
set\_hyperparam\_value, 258  
set\_hyperparams, 258  
set\_image\_in\_stack, 259  
set\_intensity\_mcmc\_sampling, 259  
set\_lbound, 259  
set\_max\_sigma, 259  
set\_mcmc\_num\_phases, 259  
set\_mcmc\_sigma\_scale, 260  
set\_min\_sigma, 260  
set\_param\_names, 260  
set\_prior, 260  
set\_rng\_seed, 260  
set\_size, 261  
set\_ubound, 261  
sigma\_scale, 267  
size, 267  
StencilVecT, 245  
theta\_in\_bounds, 261  
theta\_stack\_in\_bounds, 261  
ubound, 267  
mappel::Gauss1DsModel, 268  
bound\_theta, 274  
bounded\_theta, 274  
bounded\_theta\_stack, 274  
bounds\_epsilon, 290  
check\_image\_shape, 275  
check\_param\_shape, 275  
check\_psf\_sigma, 275



check\_size, 276  
default\_alpha\_sigma, 290  
default\_beta\_pos, 290  
default\_intensity\_kappa, 290  
default\_max\_l, 290  
default\_mean\_l, 290  
default\_pixel\_mean\_bg, 290  
default\_sigma\_pos, 291  
DefaultPriorType, 291  
DefaultSeperableInitEstimator, 291  
eta\_bg, 291  
eta\_l, 291  
eta\_sigma, 291  
eta\_x, 292  
Gauss1DsModel, 274  
get\_hyperparam\_index, 276  
get\_hyperparam\_names, 276  
get\_hyperparam\_value, 276  
get\_hyperparams, 276  
get\_image\_from\_stack, 276  
get\_lbound, 276  
get\_max\_sigma, 277  
get\_mcmc\_num\_phases, 277  
get\_mcmc\_sigma\_scale, 277  
get\_min\_sigma, 277  
get\_num\_hyperparams, 277  
get\_num\_params, 277  
get\_num\_pixels, 277  
get\_param\_names, 278  
get\_prior, 278  
get\_rng\_generator, 278  
get\_rng\_manager, 278  
get\_size, 278  
get\_size\_image\_stack, 279  
get\_stats, 279  
get\_ubound, 279  
global\_default\_mcmc\_sigma\_scale, 292  
global\_max\_mcmc\_sigma\_scale, 292  
global\_max\_psf\_sigma, 292  
global\_max\_size, 292  
global\_min\_psf\_sigma, 292  
global\_min\_size, 293  
has\_hyperparam, 279  
ImageCoordT, 272  
ImagePixelT, 272  
ImageShapeT, 272  
ImageSizeShapeT, 272  
ImageSizeVecShapeT, 273  
ImageSizeVecT, 273  
ImageSizeT, 272  
ImageStackShapeT, 273  
ImageStackT, 273  
ImageT, 273  
initial\_theta\_estimate, 279  
lbound, 293  
make\_default\_prior, 279  
make\_default\_prior\_beta\_position, 280  
make\_default\_prior\_normal\_position, 280  
make\_image, 280  
make\_image\_stack, 280  
make\_param, 280  
make\_param\_mat, 281  
make\_param\_mat\_stack, 281  
make\_param\_stack, 281  
make\_prior\_beta\_position, 281  
make\_prior\_component\_intensity, 282  
make\_prior\_component\_position\_beta, 282  
make\_prior\_component\_position\_normal, 282  
make\_prior\_component\_sigma, 282  
make\_prior\_normal\_position, 283  
make\_stencil, 283  
num\_dim, 293  
num\_hyperparams, 293  
num\_params, 293  
num\_phases, 293  
operator=, 283, 284  
ParamVecT, 273  
ParamT, 273  
pixel\_grad, 284  
pixel\_grad2, 284  
pixel\_hess, 284  
pixel\_hess\_update, 284  
pixel\_model\_value, 284  
prior, 294  
prior\_types, 294  
reflected\_theta, 285  
reflected\_theta\_stack, 285  
rename\_hyperparam, 285  
sample\_mcmc\_candidate, 285  
sample\_prior, 285, 286  
set\_background\_mcmc\_sampling, 286  
set\_bounds, 286  
set\_hyperparam\_names, 286  
set\_hyperparam\_value, 286  
set\_hyperparams, 286  
set\_image\_in\_stack, 287  
set\_intensity\_mcmc\_sampling, 287  
set\_lbound, 287  
set\_max\_sigma, 287  
set\_mcmc\_num\_phases, 287  
set\_mcmc\_sigma\_scale, 288  
set\_min\_sigma, 288  
set\_param\_names, 288  
set\_prior, 288  
set\_rng\_seed, 288  
set\_size, 289  
set\_ubound, 289  
sigma\_scale, 294

- size, [294](#)
- StencilVecT, [274](#)
- theta\_in\_bounds, [289](#)
- theta\_stack\_in\_bounds, [289](#)
- ubound, [295](#)
- mappel::Gauss1DsModel::Stencil, [822](#)
  - bg, [824](#)
  - compute\_derivatives, [824](#)
  - DXS2, [825](#)
  - DXSX, [825](#)
  - DXS, [825](#)
  - derivatives\_computed, [825](#)
  - DX, [825](#)
  - dx, [825](#)
  - Gx, [825](#)
  - l, [824](#)
  - model, [826](#)
  - operator<<, [825](#)
  - ParamT, [823](#)
  - sigma, [824](#)
  - Stencil, [823](#)
  - theta, [826](#)
  - X, [826](#)
  - x, [824](#)
- mappel::Gauss2DMAP, [295](#)
  - bound\_theta, [303](#)
  - bounded\_theta, [303](#)
  - bounded\_theta\_stack, [303](#)
  - bounds\_epsilon, [318](#)
  - check\_image\_shape, [303](#)
  - check\_param\_shape, [303](#), [304](#)
  - check\_psf\_sigma, [304](#)
  - check\_size, [304](#)
  - debug\_internal\_sum\_model\_x, [304](#)
  - debug\_internal\_sum\_model\_y, [304](#)
  - default\_alpha\_sigma, [318](#)
  - default\_beta\_pos, [319](#)
  - default\_intensity\_kappa, [319](#)
  - default\_max\_l, [319](#)
  - default\_mean\_l, [319](#)
  - default\_pixel\_mean\_bg, [319](#)
  - default\_sigma\_pos, [319](#)
  - DefaultPriorType, [320](#)
  - DefaultSeperableInitEstimator, [320](#)
  - estimator\_names, [320](#)
  - eta\_bg, [320](#)
  - eta\_l, [320](#)
  - eta\_x, [320](#)
  - eta\_y, [321](#)
  - Gauss1DSumModelT, [300](#)
  - Gauss2DMAP, [302](#), [303](#)
  - get\_hyperparam\_index, [305](#)
  - get\_hyperparam\_names, [305](#)
  - get\_hyperparam\_value, [305](#)
  - get\_hyperparams, [305](#)
  - get\_image\_from\_stack, [305](#)
  - get\_lbound, [305](#)
  - get\_mcmc\_num\_phases, [305](#)
  - get\_mcmc\_sigma\_scale, [306](#)
  - get\_num\_hyperparams, [306](#)
  - get\_num\_params, [306](#)
  - get\_num\_pixels, [306](#)
  - get\_param\_names, [306](#)
  - get\_prior, [306](#)
  - get\_psf\_sigma, [307](#)
  - get\_rng\_generator, [307](#)
  - get\_rng\_manager, [307](#)
  - get\_size, [307](#)
  - get\_size\_image\_stack, [307](#)
  - get\_stats, [308](#)
  - get\_ubound, [308](#)
  - global\_default\_mcmc\_sigma\_scale, [321](#)
  - global\_max\_mcmc\_sigma\_scale, [321](#)
  - global\_max\_psf\_sigma, [321](#)
  - global\_max\_size, [321](#)
  - global\_min\_psf\_sigma, [321](#)
  - global\_min\_size, [322](#)
  - has\_hyperparam, [308](#)
  - ImageCoordT, [300](#)
  - ImagePixelT, [300](#)
  - ImageShapeT, [300](#)
  - ImageSizeShapeT, [300](#)
  - ImageSizeVecShapeT, [300](#)
  - ImageSizeVecT, [301](#)
  - ImageSizeT, [300](#)
  - ImageStackShapeT, [301](#)
  - ImageStackT, [301](#)
  - ImageT, [301](#)
  - initial\_theta\_estimate, [308](#)
  - lbound, [322](#)
  - make\_default\_prior, [309](#)
  - make\_default\_prior\_beta\_position, [309](#)
  - make\_default\_prior\_normal\_position, [309](#)
  - make\_image, [309](#)
  - make\_image\_stack, [309](#)
  - make\_internal\_1Dsum\_estimator, [310](#)
  - make\_param, [310](#)
  - make\_param\_mat, [310](#)
  - make\_param\_mat\_stack, [310](#), [311](#)
  - make\_param\_stack, [311](#)
  - make\_prior\_beta\_position, [311](#)
  - make\_prior\_component\_intensity, [311](#)
  - make\_prior\_component\_position\_beta, [311](#)
  - make\_prior\_component\_position\_normal, [312](#)
  - make\_prior\_component\_sigma, [312](#)
  - make\_prior\_normal\_position, [312](#)
  - make\_stencil, [312](#)
  - ModelDataStackT, [301](#)

ModelDataT, 301  
name, 322  
num\_dim, 322  
num\_hyperparams, 322  
num\_params, 322  
num\_phases, 323  
operator=, 313  
ParamVecT, 302  
ParamT, 301  
pixel\_grad, 313  
pixel\_grad2, 313  
pixel\_hess, 313  
pixel\_hess\_update, 314  
pixel\_model\_value, 314  
prior, 323  
prior\_types, 323  
psf\_sigma, 323  
reflected\_theta, 314  
reflected\_theta\_stack, 314  
rename\_hyperparam, 314  
sample\_mcmc\_candidate, 314, 315  
sample\_prior, 315  
set\_background\_mcmc\_sampling, 315  
set\_bounds, 315  
set\_hyperparam\_names, 315  
set\_hyperparam\_value, 316  
set\_hyperparams, 316  
set\_image\_in\_stack, 316  
set\_intensity\_mcmc\_sampling, 316  
set\_lbound, 316  
set\_mcmc\_num\_phases, 316  
set\_mcmc\_sigma\_scale, 316  
set\_param\_names, 317  
set\_prior, 317  
set\_psf\_sigma, 317  
set\_rng\_seed, 317  
set\_size, 317  
set\_ubound, 317  
sigma\_scale, 324  
size, 324  
StencilVecT, 302  
theta\_in\_bounds, 318  
theta\_stack\_in\_bounds, 318  
ubound, 324  
update\_internal\_1Dsum\_estimators, 318  
x\_model, 325  
y\_model, 325  
mappel::Gauss2DMLE, 325  
bound\_theta, 333  
bounded\_theta, 333  
bounded\_theta\_stack, 333  
bounds\_epsilon, 348  
check\_image\_shape, 333  
check\_param\_shape, 333, 334  
check\_psf\_sigma, 334  
check\_size, 334  
debug\_internal\_sum\_model\_x, 334  
debug\_internal\_sum\_model\_y, 334  
default\_alpha\_sigma, 348  
default\_beta\_pos, 349  
default\_intensity\_kappa, 349  
default\_max\_l, 349  
default\_mean\_l, 349  
default\_pixel\_mean\_bg, 349  
default\_sigma\_pos, 349  
DefaultPriorType, 350  
DefaultSeperableInitEstimator, 350  
estimator\_names, 350  
eta\_bg, 350  
eta\_l, 350  
eta\_x, 350  
eta\_y, 351  
Gauss1DSumModelT, 330  
Gauss2DMLE, 332  
get\_hyperparam\_index, 335  
get\_hyperparam\_names, 335  
get\_hyperparam\_value, 335  
get\_hyperparams, 335  
get\_image\_from\_stack, 335  
get\_lbound, 335  
get\_mcmc\_num\_phases, 335  
get\_mcmc\_sigma\_scale, 336  
get\_num\_hyperparams, 336  
get\_num\_params, 336  
get\_num\_pixels, 336  
get\_param\_names, 336  
get\_prior, 336  
get\_psf\_sigma, 337  
get\_rng\_generator, 337  
get\_rng\_manager, 337  
get\_size, 337  
get\_size\_image\_stack, 337  
get\_stats, 338  
get\_ubound, 338  
global\_default\_mcmc\_sigma\_scale, 351  
global\_max\_mcmc\_sigma\_scale, 351  
global\_max\_psf\_sigma, 351  
global\_max\_size, 351  
global\_min\_psf\_sigma, 351  
global\_min\_size, 352  
has\_hyperparam, 338  
ImageCoordT, 330  
ImagePixelT, 330  
ImageShapeT, 330  
ImageSizeShapeT, 330  
ImageSizeVecShapeT, 330  
ImageSizeVecT, 331  
ImageSizeT, 330

ImageStackShapeT, 331  
 ImageStackT, 331  
 ImageT, 331  
 initial\_theta\_estimate, 338  
 lbound, 352  
 make\_default\_prior, 339  
 make\_default\_prior\_beta\_position, 339  
 make\_default\_prior\_normal\_position, 339  
 make\_image, 339  
 make\_image\_stack, 339  
 make\_internal\_1Dsum\_estimator, 340  
 make\_param, 340  
 make\_param\_mat, 340  
 make\_param\_mat\_stack, 340, 341  
 make\_param\_stack, 341  
 make\_prior\_beta\_position, 341  
 make\_prior\_component\_intensity, 341  
 make\_prior\_component\_position\_beta, 341  
 make\_prior\_component\_position\_normal, 342  
 make\_prior\_component\_sigma, 342  
 make\_prior\_normal\_position, 342  
 make\_stencil, 342  
 ModelDataStackT, 331  
 ModelDataT, 331  
 name, 352  
 num\_dim, 352  
 num\_hyperparams, 352  
 num\_params, 352  
 num\_phases, 353  
 operator=, 343  
 ParamVecT, 332  
 ParamT, 331  
 pixel\_grad, 343  
 pixel\_grad2, 343  
 pixel\_hess, 343  
 pixel\_hess\_update, 344  
 pixel\_model\_value, 344  
 prior, 353  
 prior\_types, 353  
 psf\_sigma, 353  
 reflected\_theta, 344  
 reflected\_theta\_stack, 344  
 rename\_hyperparam, 344  
 sample\_mcmc\_candidate, 344, 345  
 sample\_prior, 345  
 set\_background\_mcmc\_sampling, 345  
 set\_bounds, 345  
 set\_hyperparam\_names, 345  
 set\_hyperparam\_value, 346  
 set\_hyperparams, 346  
 set\_image\_in\_stack, 346  
 set\_intensity\_mcmc\_sampling, 346  
 set\_lbound, 346  
 set\_mcmc\_num\_phases, 346  
 set\_mcmc\_sigma\_scale, 346  
 set\_param\_names, 347  
 set\_prior, 347  
 set\_psf\_sigma, 347  
 set\_rng\_seed, 347  
 set\_size, 347  
 set\_ubound, 347  
 sigma\_scale, 354  
 size, 354  
 StencilVecT, 332  
 theta\_in\_bounds, 348  
 theta\_stack\_in\_bounds, 348  
 ubound, 354  
 update\_internal\_1Dsum\_estimators, 348  
 x\_model, 355  
 y\_model, 355  
 mappel::Gauss2DModel, 355  
 bound\_theta, 362  
 bounded\_theta, 362  
 bounded\_theta\_stack, 362  
 bounds\_epsilon, 378  
 check\_image\_shape, 363  
 check\_param\_shape, 363  
 check\_psf\_sigma, 363  
 check\_size, 364  
 debug\_internal\_sum\_model\_x, 364  
 debug\_internal\_sum\_model\_y, 364  
 default\_alpha\_sigma, 378  
 default\_beta\_pos, 378  
 default\_intensity\_kappa, 378  
 default\_max\_l, 378  
 default\_mean\_l, 378  
 default\_pixel\_mean\_bg, 379  
 default\_sigma\_pos, 379  
 DefaultPriorType, 379  
 DefaultSeperableInitEstimator, 379  
 eta\_bg, 379  
 eta\_l, 379  
 eta\_x, 380  
 eta\_y, 380  
 Gauss1DsumModelT, 360  
 Gauss2DModel, 362  
 get\_hyperparam\_index, 364  
 get\_hyperparam\_names, 364  
 get\_hyperparam\_value, 364  
 get\_hyperparams, 364  
 get\_image\_from\_stack, 365  
 get\_lbound, 365  
 get\_mcmc\_num\_phases, 365  
 get\_mcmc\_sigma\_scale, 365  
 get\_num\_hyperparams, 365  
 get\_num\_params, 365  
 get\_num\_pixels, 365  
 get\_param\_names, 366

get\_prior, 366  
get\_psf\_sigma, 366  
get\_rng\_generator, 366  
get\_rng\_manager, 366  
get\_size, 367  
get\_size\_image\_stack, 367  
get\_stats, 367  
get\_ubound, 367  
global\_default\_mcmc\_sigma\_scale, 380  
global\_max\_mcmc\_sigma\_scale, 380  
global\_max\_psf\_sigma, 380  
global\_max\_size, 380  
global\_min\_psf\_sigma, 381  
global\_min\_size, 381  
has\_hyperparam, 367  
ImageCoordT, 360  
ImagePixelT, 360  
ImageShapeT, 360  
ImageSizeShapeT, 360  
ImageSizeVecShapeT, 360  
ImageSizeVecT, 361  
ImageSizeT, 360  
ImageStackShapeT, 361  
ImageStackT, 361  
ImageT, 361  
initial\_theta\_estimate, 367, 368  
lbound, 381  
make\_default\_prior, 368  
make\_default\_prior\_beta\_position, 368  
make\_default\_prior\_normal\_position, 368  
make\_image, 368  
make\_image\_stack, 369  
make\_internal\_1Dsum\_estimator, 369  
make\_param, 369  
make\_param\_mat, 369  
make\_param\_mat\_stack, 370  
make\_param\_stack, 370  
make\_prior\_beta\_position, 370  
make\_prior\_component\_intensity, 370  
make\_prior\_component\_position\_beta, 371  
make\_prior\_component\_position\_normal, 371  
make\_prior\_component\_sigma, 371  
make\_prior\_normal\_position, 371  
make\_stencil, 371  
num\_dim, 381  
num\_hyperparams, 381  
num\_params, 381  
num\_phases, 382  
operator=, 372  
ParamVecT, 361  
ParamT, 361  
pixel\_grad, 372  
pixel\_grad2, 372  
pixel\_hess, 373  
pixel\_hess\_update, 373  
pixel\_model\_value, 373  
prior, 382  
prior\_types, 382  
psf\_sigma, 382  
reflected\_theta, 373  
reflected\_theta\_stack, 373  
rename\_hyperparam, 373  
sample\_mcmc\_candidate, 374  
sample\_prior, 374  
set\_background\_mcmc\_sampling, 374  
set\_bounds, 374  
set\_hyperparam\_names, 375  
set\_hyperparam\_value, 375  
set\_hyperparams, 375  
set\_image\_in\_stack, 375  
set\_intensity\_mcmc\_sampling, 375  
set\_lbound, 375  
set\_mcmc\_num\_phases, 376  
set\_mcmc\_sigma\_scale, 376  
set\_param\_names, 376  
set\_prior, 376  
set\_psf\_sigma, 376  
set\_rng\_seed, 377  
set\_size, 377  
set\_ubound, 377  
sigma\_scale, 383  
size, 383  
StencilVecT, 361  
theta\_in\_bounds, 377  
theta\_stack\_in\_bounds, 377  
ubound, 383  
update\_internal\_1Dsum\_estimators, 377  
x\_model, 384  
y\_model, 384  
mappel::Gauss2DModel::Stencil, 826  
bg, 828  
compute\_derivatives, 828  
DXS, 829  
DYS, 830  
derivatives\_computed, 829  
DX, 829  
dx, 829  
DY, 830  
dy, 829  
Gx, 830  
Gy, 830  
I, 828  
model, 830  
operator<<, 829  
ParamT, 827  
Stencil, 827  
theta, 830  
X, 830

- x, [828](#)
- Y, [830](#)
- y, [828](#)
- mappel::Gauss2DsMAP, [384](#)
  - bound\_theta, [392](#)
  - bounded\_theta, [392](#)
  - bounded\_theta\_stack, [392](#)
  - bounds\_epsilon, [410](#)
  - check\_image\_shape, [393](#)
  - check\_param\_shape, [393](#)
  - check\_psf\_sigma, [393](#)
  - check\_size, [394](#)
  - compute\_max\_sigma\_ratio, [394](#)
  - debug\_internal\_sum\_model\_x, [394](#)
  - debug\_internal\_sum\_model\_y, [394](#)
  - default\_alpha\_sigma, [410](#)
  - default\_beta\_pos, [410](#)
  - default\_intensity\_kappa, [410](#)
  - default\_max\_l, [410](#)
  - default\_mean\_l, [410](#)
  - default\_pixel\_mean\_bg, [410](#)
  - default\_sigma\_pos, [411](#)
  - DefaultPriorType, [411](#)
  - DefaultSeperableInitEstimator, [411](#)
  - estimator\_names, [411](#)
  - eta\_bg, [411](#)
  - eta\_l, [411](#)
  - eta\_sigma, [412](#)
  - eta\_x, [412](#)
  - eta\_y, [412](#)
  - Gauss1DSumModelT, [389](#)
  - Gauss2DsMAP, [391](#), [392](#)
  - get\_hyperparam\_index, [394](#)
  - get\_hyperparam\_names, [394](#)
  - get\_hyperparam\_value, [395](#)
  - get\_hyperparams, [395](#)
  - get\_image\_from\_stack, [395](#)
  - get\_lbound, [395](#)
  - get\_max\_sigma, [395](#)
  - get\_max\_sigma\_ratio, [396](#)
  - get\_mcmc\_num\_phases, [396](#)
  - get\_mcmc\_sigma\_scale, [396](#)
  - get\_min\_sigma, [396](#)
  - get\_num\_hyperparams, [396](#)
  - get\_num\_params, [396](#)
  - get\_num\_pixels, [397](#)
  - get\_param\_names, [397](#)
  - get\_prior, [397](#)
  - get\_rng\_generator, [397](#)
  - get\_rng\_manager, [397](#)
  - get\_size, [397](#), [398](#)
  - get\_size\_image\_stack, [398](#)
  - get\_stats, [398](#)
  - get\_ubound, [398](#)
  - global\_default\_mcmc\_sigma\_scale, [412](#)
  - global\_max\_mcmc\_sigma\_scale, [412](#)
  - global\_max\_psf\_sigma, [413](#)
  - global\_max\_size, [413](#)
  - global\_min\_psf\_sigma, [413](#)
  - global\_min\_size, [413](#)
  - has\_hyperparam, [398](#)
  - ImageCoordT, [389](#)
  - ImagePixelT, [389](#)
  - ImageShapeT, [389](#)
  - ImageSizeShapeT, [390](#)
  - ImageSizeVecShapeT, [390](#)
  - ImageSizeVecT, [390](#)
  - ImageSizeT, [390](#)
  - ImageStackShapeT, [390](#)
  - ImageStackT, [390](#)
  - ImageT, [390](#)
  - initial\_theta\_estimate, [398](#), [399](#)
  - lbound, [413](#)
  - make\_default\_prior, [399](#)
  - make\_default\_prior\_beta\_position, [399](#)
  - make\_default\_prior\_normal\_position, [399](#)
  - make\_image, [399](#)
  - make\_image\_stack, [400](#)
  - make\_internal\_1Dsum\_estimator, [400](#)
  - make\_param, [400](#)
  - make\_param\_mat, [400](#)
  - make\_param\_mat\_stack, [401](#)
  - make\_param\_stack, [401](#)
  - make\_prior\_beta\_position, [401](#)
  - make\_prior\_component\_intensity, [401](#)
  - make\_prior\_component\_position\_beta, [402](#)
  - make\_prior\_component\_position\_normal, [402](#)
  - make\_prior\_component\_sigma, [402](#)
  - make\_prior\_normal\_position, [402](#)
  - make\_stencil, [403](#)
  - min\_sigma, [413](#)
  - ModelDataStackT, [391](#)
  - ModelDataT, [391](#)
  - name, [414](#)
  - num\_dim, [414](#)
  - num\_hyperparams, [414](#)
  - num\_params, [414](#)
  - num\_phases, [414](#)
  - operator=, [403](#)
  - ParamVecT, [391](#)
  - ParamT, [391](#)
  - pixel\_grad, [404](#)
  - pixel\_grad2, [404](#)
  - pixel\_hess, [404](#)
  - pixel\_hess\_update, [404](#)
  - pixel\_model\_value, [404](#)
  - prior, [415](#)
  - prior\_types, [415](#)

- reflected\_theta, 405
- reflected\_theta\_stack, 405
- rename\_hyperparam, 405
- sample\_mcmc\_candidate, 405
- sample\_prior, 405, 406
- set\_background\_mcmc\_sampling, 406
- set\_bounds, 406
- set\_hyperparam\_names, 406
- set\_hyperparam\_value, 406
- set\_hyperparams, 406
- set\_image\_in\_stack, 407
- set\_intensity\_mcmc\_sampling, 407
- set\_lbound, 407
- set\_max\_sigma, 407
- set\_max\_sigma\_ratio, 407
- set\_mcmc\_num\_phases, 407
- set\_mcmc\_sigma\_scale, 408
- set\_min\_sigma, 408
- set\_param\_names, 408
- set\_prior, 408
- set\_rng\_seed, 408
- set\_size, 408
- set\_ubound, 409
- sigma\_scale, 415
- size, 415
- StencilVecT, 391
- theta\_in\_bounds, 409
- theta\_stack\_in\_bounds, 409
- ubound, 416
- update\_internal\_1Dsum\_estimators, 409
- x\_model, 416
- y\_model, 416
- mappel::Gauss2DsMLE, 417
  - bound\_theta, 424
  - bounded\_theta, 424
  - bounded\_theta\_stack, 425
  - bounds\_epsilon, 442
  - check\_image\_shape, 425
  - check\_param\_shape, 425
  - check\_psf\_sigma, 425, 426
  - check\_size, 426
  - compute\_max\_sigma\_ratio, 426
  - debug\_internal\_sum\_model\_x, 426
  - debug\_internal\_sum\_model\_y, 426
  - default\_alpha\_sigma, 442
  - default\_beta\_pos, 442
  - default\_intensity\_kappa, 442
  - default\_max\_l, 442
  - default\_mean\_l, 442
  - default\_pixel\_mean\_bg, 443
  - default\_sigma\_pos, 443
  - DefaultPriorType, 443
  - DefaultSeperableInitEstimator, 443
  - estimator\_names, 443
  - eta\_bg, 443
  - eta\_l, 444
  - eta\_sigma, 444
  - eta\_x, 444
  - eta\_y, 444
  - Gauss1DSumModelT, 421
  - Gauss2DsMLE, 424
  - get\_hyperparam\_index, 427
  - get\_hyperparam\_names, 427
  - get\_hyperparam\_value, 427
  - get\_hyperparams, 427
  - get\_image\_from\_stack, 427
  - get\_lbound, 427
  - get\_max\_sigma, 427, 428
  - get\_max\_sigma\_ratio, 428
  - get\_mcmc\_num\_phases, 428
  - get\_mcmc\_sigma\_scale, 428
  - get\_min\_sigma, 428
  - get\_num\_hyperparams, 428
  - get\_num\_params, 429
  - get\_num\_pixels, 429
  - get\_param\_names, 429
  - get\_prior, 429
  - get\_rng\_generator, 429
  - get\_rng\_manager, 429
  - get\_size, 430
  - get\_size\_image\_stack, 430
  - get\_stats, 430
  - get\_ubound, 430
  - global\_default\_mcmc\_sigma\_scale, 444
  - global\_max\_mcmc\_sigma\_scale, 445
  - global\_max\_psf\_sigma, 445
  - global\_max\_size, 445
  - global\_min\_psf\_sigma, 445
  - global\_min\_size, 445
  - has\_hyperparam, 430
  - ImageCoordT, 421
  - ImagePixelT, 422
  - ImageShapeT, 422
  - ImageSizeShapeT, 422
  - ImageSizeVecShapeT, 422
  - ImageSizeVecT, 422
  - ImageSizeT, 422
  - ImageStackShapeT, 422
  - ImageStackT, 423
  - ImageT, 423
  - initial\_theta\_estimate, 430, 431
  - lbound, 445
  - make\_default\_prior, 431
  - make\_default\_prior\_beta\_position, 431
  - make\_default\_prior\_normal\_position, 431
  - make\_image, 432
  - make\_image\_stack, 432
  - make\_internal\_1Dsum\_estimator, 432

[make\\_param](#), 432  
[make\\_param\\_mat](#), 433  
[make\\_param\\_mat\\_stack](#), 433  
[make\\_param\\_stack](#), 433  
[make\\_prior\\_beta\\_position](#), 433  
[make\\_prior\\_component\\_intensity](#), 434  
[make\\_prior\\_component\\_position\\_beta](#), 434  
[make\\_prior\\_component\\_position\\_normal](#), 434  
[make\\_prior\\_component\\_sigma](#), 434  
[make\\_prior\\_normal\\_position](#), 435  
[make\\_stencil](#), 435  
[min\\_sigma](#), 446  
[ModelDataStackT](#), 423  
[ModelDataT](#), 423  
[name](#), 446  
[num\\_dim](#), 446  
[num\\_hyperparams](#), 446  
[num\\_params](#), 446  
[num\\_phases](#), 446  
[operator=](#), 435, 436  
[ParamVecT](#), 423  
[ParamT](#), 423  
[pixel\\_grad](#), 436  
[pixel\\_grad2](#), 436  
[pixel\\_hess](#), 436  
[pixel\\_hess\\_update](#), 436  
[pixel\\_model\\_value](#), 437  
[prior](#), 447  
[prior\\_types](#), 447  
[reflected\\_theta](#), 437  
[reflected\\_theta\\_stack](#), 437  
[rename\\_hyperparam](#), 437  
[sample\\_mcmc\\_candidate](#), 437, 438  
[sample\\_prior](#), 438  
[set\\_background\\_mcmc\\_sampling](#), 438  
[set\\_bounds](#), 438  
[set\\_hyperparam\\_names](#), 438  
[set\\_hyperparam\\_value](#), 439  
[set\\_hyperparams](#), 439  
[set\\_image\\_in\\_stack](#), 439  
[set\\_intensity\\_mcmc\\_sampling](#), 439  
[set\\_lbound](#), 439  
[set\\_max\\_sigma](#), 439  
[set\\_max\\_sigma\\_ratio](#), 439  
[set\\_mcmc\\_num\\_phases](#), 440  
[set\\_mcmc\\_sigma\\_scale](#), 440  
[set\\_min\\_sigma](#), 440  
[set\\_param\\_names](#), 440  
[set\\_prior](#), 440  
[set\\_rng\\_seed](#), 441  
[set\\_size](#), 441  
[set\\_ubound](#), 441  
[sigma\\_scale](#), 447  
[size](#), 447  
[StencilVecT](#), 423  
[theta\\_in\\_bounds](#), 441  
[theta\\_stack\\_in\\_bounds](#), 441  
[ubound](#), 448  
[update\\_internal\\_1Dsum\\_estimators](#), 441  
[x\\_model](#), 448  
[y\\_model](#), 448  
[mappel::Gauss2DsModel](#), 449  
[bound\\_theta](#), 456  
[bounded\\_theta](#), 456  
[bounded\\_theta\\_stack](#), 456  
[bounds\\_epsilon](#), 473  
[check\\_image\\_shape](#), 457  
[check\\_param\\_shape](#), 457  
[check\\_psf\\_sigma](#), 457  
[check\\_size](#), 458  
[compute\\_max\\_sigma\\_ratio](#), 458  
[debug\\_internal\\_sum\\_model\\_x](#), 458  
[debug\\_internal\\_sum\\_model\\_y](#), 458  
[default\\_alpha\\_sigma](#), 473  
[default\\_beta\\_pos](#), 473  
[default\\_intensity\\_kappa](#), 473  
[default\\_max\\_l](#), 474  
[default\\_mean\\_l](#), 474  
[default\\_pixel\\_mean\\_bg](#), 474  
[default\\_sigma\\_pos](#), 474  
[DefaultPriorType](#), 474  
[DefaultSeperableInitEstimator](#), 474  
[eta\\_bg](#), 475  
[eta\\_l](#), 475  
[eta\\_sigma](#), 475  
[eta\\_x](#), 475  
[eta\\_y](#), 475  
[Gauss1DSumModelT](#), 454  
[Gauss2DsModel](#), 456  
[get\\_hyperparam\\_index](#), 458  
[get\\_hyperparam\\_names](#), 458  
[get\\_hyperparam\\_value](#), 459  
[get\\_hyperparams](#), 459  
[get\\_image\\_from\\_stack](#), 459  
[get\\_lbound](#), 459  
[get\\_max\\_sigma](#), 459  
[get\\_max\\_sigma\\_ratio](#), 460  
[get\\_mcmc\\_num\\_phases](#), 460  
[get\\_mcmc\\_sigma\\_scale](#), 460  
[get\\_min\\_sigma](#), 460  
[get\\_num\\_hyperparams](#), 460  
[get\\_num\\_params](#), 460  
[get\\_num\\_pixels](#), 460  
[get\\_param\\_names](#), 461  
[get\\_prior](#), 461  
[get\\_rng\\_generator](#), 461  
[get\\_rng\\_manager](#), 461  
[get\\_size](#), 461



get\_size\_image\_stack, 462  
get\_stats, 462  
get\_ubound, 462  
global\_default\_mcmc\_sigma\_scale, 476  
global\_max\_mcmc\_sigma\_scale, 476  
global\_max\_psf\_sigma, 476  
global\_max\_size, 476  
global\_min\_psf\_sigma, 476  
global\_min\_size, 476  
has\_hyperparam, 462  
ImageCoordT, 454  
ImagePixelT, 454  
ImageShapeT, 454  
ImageSizeShapeT, 454  
ImageSizeVecShapeT, 454  
ImageSizeVecT, 455  
ImageSizeT, 454  
ImageStackShapeT, 455  
ImageStackT, 455  
ImageT, 455  
initial\_theta\_estimate, 462, 463  
lbound, 477  
make\_default\_prior, 463  
make\_default\_prior\_beta\_position, 463  
make\_default\_prior\_normal\_position, 463  
make\_image, 463  
make\_image\_stack, 464  
make\_internal\_1Dsum\_estimator, 464  
make\_param, 464  
make\_param\_mat, 464  
make\_param\_mat\_stack, 465  
make\_param\_stack, 465  
make\_prior\_beta\_position, 465  
make\_prior\_component\_intensity, 465  
make\_prior\_component\_position\_beta, 466  
make\_prior\_component\_position\_normal, 466  
make\_prior\_component\_sigma, 466  
make\_prior\_normal\_position, 466  
make\_stencil, 466  
min\_sigma, 477  
num\_dim, 477  
num\_hyperparams, 477  
num\_params, 477  
num\_phases, 478  
operator=, 467  
ParamVecT, 455  
ParamT, 455  
pixel\_grad, 467  
pixel\_grad2, 467  
pixel\_hess, 468  
pixel\_hess\_update, 468  
pixel\_model\_value, 468  
prior, 478  
prior\_types, 478  
reflected\_theta, 468  
reflected\_theta\_stack, 468  
rename\_hyperparam, 469  
sample\_mcmc\_candidate, 469  
sample\_prior, 469  
set\_background\_mcmc\_sampling, 469  
set\_bounds, 470  
set\_hyperparam\_names, 470  
set\_hyperparam\_value, 470  
set\_hyperparams, 470  
set\_image\_in\_stack, 470  
set\_intensity\_mcmc\_sampling, 470  
set\_lbound, 471  
set\_max\_sigma, 471  
set\_max\_sigma\_ratio, 471  
set\_mcmc\_num\_phases, 471  
set\_mcmc\_sigma\_scale, 471  
set\_min\_sigma, 471  
set\_param\_names, 472  
set\_prior, 472  
set\_rng\_seed, 472  
set\_size, 472  
set\_ubound, 472  
sigma\_scale, 478  
size, 479  
StencilVecT, 455  
theta\_in\_bounds, 472  
theta\_stack\_in\_bounds, 473  
ubound, 479  
update\_internal\_1Dsum\_estimators, 473  
x\_model, 479  
y\_model, 480  
mappel::Gauss2DsModel::Stencil, 831  
bg, 832  
compute\_derivatives, 832  
DXS2, 834  
DXSX, 835  
DXS, 834  
DYS2, 835  
DYSY, 835  
DYS, 835  
derivatives\_computed, 834  
DX, 834  
dx, 834  
DY, 835  
dy, 835  
Gx, 835  
Gy, 836  
I, 833  
model, 836  
operator<<, 834  
ParamT, 832  
sigma\_ratio, 833  
sigmaX, 833

- sigmaY, [833](#)
- Stencil, [832](#)
- theta, [836](#)
- X, [836](#)
- x, [833](#)
- Y, [836](#)
- y, [833](#)
- mappel::Gauss2DsxyMAP, [480](#)
  - bound\_theta, [486](#)
  - bounded\_theta, [486](#)
  - bounded\_theta\_stack, [487](#)
  - bounds\_epsilon, [501](#)
  - check\_image\_shape, [487](#)
  - check\_param\_shape, [487](#)
  - check\_psf\_sigma, [487](#), [488](#)
  - check\_size, [488](#)
  - default\_alpha\_sigma, [501](#)
  - default\_beta\_pos, [501](#)
  - default\_intensity\_kappa, [501](#)
  - default\_max\_l, [501](#)
  - default\_mean\_l, [502](#)
  - default\_pixel\_mean\_bg, [502](#)
  - default\_sigma\_pos, [502](#)
  - DefaultPriorType, [502](#)
  - DefaultSeperableInitEstimator, [502](#)
  - estimator\_names, [502](#)
  - eta\_bg, [503](#)
  - eta\_l, [503](#)
  - eta\_x, [503](#)
  - Gauss2DsxyMAP, [486](#)
  - get\_hyperparam\_index, [488](#)
  - get\_hyperparam\_names, [488](#)
  - get\_hyperparam\_value, [488](#)
  - get\_hyperparams, [488](#)
  - get\_image\_from\_stack, [489](#)
  - get\_lbound, [489](#)
  - get\_mcmc\_num\_phases, [489](#)
  - get\_mcmc\_sigma\_scale, [489](#)
  - get\_num\_hyperparams, [489](#)
  - get\_num\_params, [489](#)
  - get\_num\_pixels, [489](#)
  - get\_param\_names, [490](#)
  - get\_prior, [490](#)
  - get\_psf\_sigma, [490](#)
  - get\_rng\_generator, [490](#)
  - get\_rng\_manager, [490](#)
  - get\_size, [491](#)
  - get\_size\_image\_stack, [491](#)
  - get\_stats, [491](#)
  - get\_ubound, [491](#)
  - global\_default\_mcmc\_sigma\_scale, [503](#)
  - global\_max\_mcmc\_sigma\_scale, [503](#)
  - global\_max\_psf\_sigma, [503](#)
  - global\_max\_size, [504](#)
  - global\_min\_psf\_sigma, [504](#)
  - global\_min\_size, [504](#)
  - has\_hyperparam, [491](#)
  - ImageCoordT, [484](#)
  - ImagePixelT, [484](#)
  - ImageShapeT, [484](#)
  - ImageSizeShapeT, [484](#)
  - ImageSizeVecShapeT, [485](#)
  - ImageSizeVecT, [485](#)
  - ImageSizeT, [485](#)
  - ImageStackShapeT, [485](#)
  - ImageStackT, [485](#)
  - ImageT, [485](#)
  - initial\_theta\_estimate, [491](#), [492](#)
  - lbound, [504](#)
  - make\_default\_prior, [492](#)
  - make\_default\_prior\_beta\_position, [492](#)
  - make\_default\_prior\_normal\_position, [492](#)
  - make\_image, [492](#)
  - make\_image\_stack, [492](#)
  - make\_param, [493](#)
  - make\_param\_mat, [493](#)
  - make\_param\_mat\_stack, [493](#)
  - make\_param\_stack, [493](#), [494](#)
  - make\_prior\_beta\_position, [494](#)
  - make\_prior\_component\_intensity, [494](#)
  - make\_prior\_component\_position\_beta, [494](#)
  - make\_prior\_component\_position\_normal, [494](#)
  - make\_prior\_component\_sigma, [495](#)
  - make\_prior\_normal\_position, [495](#)
  - make\_stencil, [495](#)
  - ModelDataStackT, [485](#)
  - ModelDataT, [486](#)
  - name, [504](#)
  - num\_dim, [504](#)
  - num\_hyperparams, [505](#)
  - num\_params, [505](#)
  - num\_phases, [505](#)
  - ParamVecT, [486](#)
  - ParamT, [486](#)
  - pixel\_grad, [496](#)
  - pixel\_grad2, [496](#)
  - pixel\_hess, [496](#)
  - pixel\_hess\_update, [496](#)
  - pixel\_model\_value, [496](#)
  - prior, [505](#)
  - prior\_types, [505](#)
  - psf\_sigma, [506](#)
  - reflected\_theta, [496](#)
  - reflected\_theta\_stack, [497](#)
  - rename\_hyperparam, [497](#)
  - sample\_mcmc\_candidate, [497](#)
  - sample\_prior, [497](#)
  - set\_background\_mcmc\_sampling, [498](#)

- set\_bounds, 498
- set\_hyperparam\_names, 498
- set\_hyperparam\_value, 498
- set\_hyperparams, 498
- set\_image\_in\_stack, 498
- set\_intensity\_mcmc\_sampling, 499
- set\_lbound, 499
- set\_mcmc\_num\_phases, 499
- set\_mcmc\_sigma\_scale, 499
- set\_param\_names, 499
- set\_prior, 499
- set\_psf\_sigma, 500
- set\_rng\_seed, 500
- set\_size, 500
- set\_ubound, 500
- sigma\_scale, 506
- size, 506
- StencilVecT, 486
- theta\_in\_bounds, 500
- theta\_stack\_in\_bounds, 501
- ubound, 506
- mappel::Gauss2DsxyModel, 507
  - bound\_theta, 513
  - bounded\_theta, 513
  - bounded\_theta\_stack, 513
  - bounds\_epsilon, 525
  - check\_image\_shape, 513
  - check\_param\_shape, 513, 514
  - check\_psf\_sigma, 514
  - check\_size, 514
  - compute\_max\_sigma\_ratio, 514
  - default\_alpha\_sigma, 525
  - default\_beta\_pos, 525
  - default\_intensity\_kappa, 526
  - default\_max\_l, 526
  - default\_mean\_l, 526
  - default\_pixel\_mean\_bg, 526
  - default\_sigma\_pos, 526
  - DefaultSeperableInitEstimator, 526
  - Gauss2DsxyModel, 513
  - get\_hyperparam\_index, 515
  - get\_hyperparam\_names, 515
  - get\_hyperparam\_value, 515
  - get\_hyperparams, 515
  - get\_image\_from\_stack, 515
  - get\_lbound, 515
  - get\_max\_sigma, 515, 516
  - get\_max\_sigma\_ratio, 516
  - get\_min\_sigma, 516
  - get\_num\_hyperparams, 516
  - get\_num\_params, 516
  - get\_num\_pixels, 516
  - get\_param\_names, 516
  - get\_prior, 517
  - get\_rng\_generator, 517
  - get\_rng\_manager, 517
  - get\_size, 517
  - get\_size\_image\_stack, 517
  - get\_stats, 518
  - get\_ubound, 518
  - global\_max\_psf\_sigma, 527
  - global\_max\_size, 527
  - global\_min\_psf\_sigma, 527
  - global\_min\_size, 527
  - has\_hyperparam, 518
  - ImageCoordT, 511
  - ImagePixelT, 511
  - ImageShapeT, 511
  - ImageSizeShapeT, 511
  - ImageSizeVecShapeT, 511
  - ImageSizeVecT, 512
  - ImageSizeT, 511
  - ImageStackShapeT, 512
  - ImageStackT, 512
  - ImageT, 512
  - initial\_theta\_estimate, 518
  - lbound, 527
  - make\_default\_prior, 518
  - make\_image, 518
  - make\_image\_stack, 518
  - make\_param, 519
  - make\_param\_mat, 519
  - make\_param\_mat\_stack, 519
  - make\_param\_stack, 519, 520
  - make\_prior\_beta\_position, 520
  - make\_prior\_component\_intensity, 520
  - make\_prior\_component\_position\_beta, 520
  - make\_prior\_component\_position\_normal, 520
  - make\_prior\_component\_sigma, 521
  - make\_prior\_normal\_position, 521
  - make\_stencil, 521
  - mcmc\_candidate\_eta\_sigma, 528
  - mcmc\_candidate\_eta\_y, 528
  - min\_sigma, 528
  - num\_dim, 528
  - num\_hyperparams, 528
  - num\_params, 528
  - ParamVecT, 512
  - ParamT, 512
  - pixel\_grad, 522
  - pixel\_grad2, 522
  - pixel\_hess, 522
  - pixel\_hess\_update, 522
  - pixel\_model\_value, 522
  - prior, 529
  - reflected\_theta, 522
  - reflected\_theta\_stack, 522
  - rename\_hyperparam, 523

- sample\_mcmc\_candidate, 523
- sample\_prior, 523
- set\_bounds, 523
- set\_hyperparam\_names, 523
- set\_hyperparam\_value, 523
- set\_hyperparams, 524
- set\_image\_in\_stack, 524
- set\_lbound, 524
- set\_max\_sigma, 524
- set\_max\_sigma\_ratio, 524
- set\_min\_sigma, 524
- set\_param\_names, 524
- set\_prior, 524
- set\_rng\_seed, 524
- set\_size, 524
- set\_ubound, 525
- size, 529
- StencilVecT, 512
- theta\_in\_bounds, 525
- theta\_stack\_in\_bounds, 525
- ubound, 530
- update\_internal\_1D\_estimators, 525
- x\_model, 530
- y\_model, 530
- mappel::Gauss2DsxyModel::Stencil, 817
  - bg, 819
  - compute\_derivatives, 819
  - DXS2, 820
  - DXSX, 820
  - DXS, 820
  - DYS2, 821
  - DYSX, 821
  - DYSY, 821
  - DYS, 821
  - derivatives\_computed, 820
  - DX, 820
  - dx, 820
  - DY, 821
  - dy, 821
  - Gx, 821
  - Gy, 821
  - l, 819
  - model, 821
  - operator<<, 820
  - ParamT, 818
  - sigmaX, 819
  - sigmaY, 819
  - Stencil, 819
  - theta, 822
  - X, 822
  - x, 819
  - Y, 822
  - y, 820
- mappel::ImageFormat1DBase, 543
  - check\_image\_shape, 546
  - check\_size, 546
  - get\_image\_from\_stack, 546
  - get\_num\_pixels, 547
  - get\_size, 547
  - get\_size\_image\_stack, 547
  - get\_stats, 547
  - global\_max\_size, 548
  - global\_min\_size, 548
  - ImageCoordT, 545
  - ImageFormat1DBase, 546
  - ImagePixelT, 545
  - ImageShapeT, 545
  - ImageSizeShapeT, 545
  - ImageSizeVecShapeT, 545
  - ImageSizeVecT, 545
  - ImageSizeT, 545
  - ImageStackShapeT, 545
  - ImageStackT, 545
  - ImageT, 546
  - make\_image, 547
  - make\_image\_stack, 547
  - num\_dim, 548
  - set\_image\_in\_stack, 548
  - set\_size, 548
  - size, 548
- mappel::ImageFormat2DBase, 549
  - check\_image\_shape, 553
  - check\_size, 553
  - get\_image\_from\_stack, 553
  - get\_num\_pixels, 553
  - get\_size, 553
  - get\_size\_image\_stack, 554
  - get\_stats, 554
  - global\_max\_size, 555
  - global\_min\_size, 555
  - ImageCoordT, 551
  - ImageFormat2DBase, 552
  - ImagePixelT, 551
  - ImageShapeT, 551
  - ImageSizeShapeT, 551
  - ImageSizeVecShapeT, 551
  - ImageSizeVecT, 551
  - ImageSizeT, 551
  - ImageStackShapeT, 552
  - ImageStackT, 552
  - ImageT, 552
  - make\_image, 554
  - make\_image\_stack, 554
  - num\_dim, 555
  - operator=, 554
  - set\_image\_in\_stack, 554
  - set\_size, 555
  - size, 555

mappel::LogicalError, 577  
  LogicalError, 577  
mappel::MAPEstimator, 577  
  bound\_theta, 580  
  bounded\_theta, 580  
  bounded\_theta\_stack, 581  
  bounds\_epsilon, 589  
  check\_param\_shape, 581  
  check\_psf\_sigma, 581  
  default\_alpha\_sigma, 589  
  default\_beta\_pos, 589  
  default\_intensity\_kappa, 589  
  default\_max\_l, 589  
  default\_mean\_l, 590  
  default\_pixel\_mean\_bg, 590  
  default\_sigma\_pos, 590  
  DefaultSeperableInitEstimator, 590  
  get\_hyperparam\_index, 581  
  get\_hyperparam\_names, 582  
  get\_hyperparam\_value, 582  
  get\_hyperparams, 582  
  get\_lbound, 582  
  get\_num\_hyperparams, 582  
  get\_num\_params, 582  
  get\_param\_names, 582  
  get\_prior, 583  
  get\_rng\_generator, 583  
  get\_rng\_manager, 583  
  get\_stats, 583  
  get\_ubound, 583  
  global\_max\_psf\_sigma, 590  
  global\_min\_psf\_sigma, 590  
  has\_hyperparam, 583  
  lbound, 591  
  MAPEstimator, 580  
  make\_param, 584  
  make\_param\_mat, 584  
  make\_param\_mat\_stack, 584  
  make\_param\_stack, 584, 585  
  make\_prior\_component\_intensity, 585  
  make\_prior\_component\_position\_beta, 585  
  make\_prior\_component\_position\_normal, 585  
  make\_prior\_component\_sigma, 585  
  num\_hyperparams, 591  
  num\_params, 591  
  operator=, 586  
  ParamVecT, 580  
  ParamT, 580  
  prior, 591  
  reflected\_theta, 586  
  reflected\_theta\_stack, 586  
  rename\_hyperparam, 586  
  sample\_prior, 586, 587  
  set\_bounds, 587  
  set\_hyperparam\_names, 587  
  set\_hyperparam\_value, 587  
  set\_hyperparams, 587  
  set\_lbound, 587  
  set\_param\_names, 588  
  set\_prior, 588  
  set\_rng\_seed, 588  
  set\_ubound, 588  
  theta\_in\_bounds, 588  
  theta\_stack\_in\_bounds, 589  
  ubound, 591  
mappel::MCMCAdaptor1Ds, 619  
  bound\_theta, 623  
  bounded\_theta, 623  
  bounded\_theta\_stack, 623  
  bounds\_epsilon, 633  
  check\_param\_shape, 623  
  check\_psf\_sigma, 624  
  default\_alpha\_sigma, 633  
  default\_beta\_pos, 633  
  default\_intensity\_kappa, 633  
  default\_max\_l, 633  
  default\_mean\_l, 633  
  default\_pixel\_mean\_bg, 633  
  default\_sigma\_pos, 634  
  DefaultSeperableInitEstimator, 634  
  eta\_bg, 634  
  eta\_l, 634  
  eta\_sigma, 634  
  eta\_x, 635  
  get\_hyperparam\_index, 624  
  get\_hyperparam\_names, 624  
  get\_hyperparam\_value, 624  
  get\_hyperparams, 624  
  get\_lbound, 625  
  get\_mcmc\_num\_phases, 625  
  get\_mcmc\_sigma\_scale, 625  
  get\_num\_hyperparams, 625  
  get\_num\_params, 625  
  get\_param\_names, 625  
  get\_prior, 625, 626  
  get\_rng\_generator, 626  
  get\_rng\_manager, 626  
  get\_stats, 626  
  get\_ubound, 626  
  global\_default\_mcmc\_sigma\_scale, 635  
  global\_max\_mcmc\_sigma\_scale, 635  
  global\_max\_psf\_sigma, 635  
  global\_min\_psf\_sigma, 635  
  has\_hyperparam, 626  
  lbound, 635  
  MCMCAdaptor1Ds, 622, 623  
  make\_param, 626, 627  
  make\_param\_mat, 627

make\_param\_mat\_stack, 627  
make\_param\_stack, 627  
make\_prior\_component\_intensity, 628  
make\_prior\_component\_position\_beta, 628  
make\_prior\_component\_position\_normal, 628  
make\_prior\_component\_sigma, 628  
num\_hyperparams, 636  
num\_params, 636  
num\_phases, 636  
operator=, 628, 629  
ParamVecT, 622  
ParamT, 622  
prior, 636  
reflected\_theta, 629  
reflected\_theta\_stack, 629  
rename\_hyperparam, 629  
sample\_mcmc\_candidate, 629  
sample\_prior, 630  
set\_background\_mcmc\_sampling, 630  
set\_bounds, 630  
set\_hyperparam\_names, 630  
set\_hyperparam\_value, 630  
set\_hyperparams, 631  
set\_intensity\_mcmc\_sampling, 631  
set\_lbound, 631  
set\_mcmc\_num\_phases, 631  
set\_mcmc\_sigma\_scale, 631  
set\_param\_names, 631  
set\_prior, 632  
set\_rng\_seed, 632  
set\_ubound, 632  
sigma\_scale, 636  
theta\_in\_bounds, 632  
theta\_stack\_in\_bounds, 632  
ubound, 637  
mappel::MCMCAdaptor1D, 601  
bound\_theta, 605  
bounded\_theta, 605  
bounded\_theta\_stack, 605  
bounds\_epsilon, 615  
check\_param\_shape, 605, 606  
check\_psf\_sigma, 606  
default\_alpha\_sigma, 615  
default\_beta\_pos, 615  
default\_intensity\_kappa, 615  
default\_max\_l, 615  
default\_mean\_l, 616  
default\_pixel\_mean\_bg, 616  
default\_sigma\_pos, 616  
DefaultSeperableInitEstimator, 616  
eta\_bg, 616  
eta\_l, 616  
eta\_x, 617  
get\_hyperparam\_index, 606  
get\_hyperparam\_names, 606  
get\_hyperparam\_value, 606  
get\_hyperparams, 607  
get\_lbound, 607  
get\_mcmc\_num\_phases, 607  
get\_mcmc\_sigma\_scale, 607  
get\_num\_hyperparams, 607  
get\_num\_params, 607  
get\_param\_names, 607  
get\_prior, 608  
get\_rng\_generator, 608  
get\_rng\_manager, 608  
get\_stats, 608  
get\_ubound, 608  
global\_default\_mcmc\_sigma\_scale, 617  
global\_max\_mcmc\_sigma\_scale, 617  
global\_max\_psf\_sigma, 617  
global\_min\_psf\_sigma, 617  
has\_hyperparam, 608  
lbound, 617  
MCMCAdaptor1D, 604, 605  
make\_param, 609  
make\_param\_mat, 609  
make\_param\_mat\_stack, 609  
make\_param\_stack, 609, 610  
make\_prior\_component\_intensity, 610  
make\_prior\_component\_position\_beta, 610  
make\_prior\_component\_position\_normal, 610  
make\_prior\_component\_sigma, 610  
num\_hyperparams, 618  
num\_params, 618  
num\_phases, 618  
operator=, 611  
ParamVecT, 604  
ParamT, 604  
prior, 618  
reflected\_theta, 611  
reflected\_theta\_stack, 611  
rename\_hyperparam, 611  
sample\_mcmc\_candidate, 611, 612  
sample\_prior, 612  
set\_background\_mcmc\_sampling, 612  
set\_bounds, 612  
set\_hyperparam\_names, 612  
set\_hyperparam\_value, 613  
set\_hyperparams, 613  
set\_intensity\_mcmc\_sampling, 613  
set\_lbound, 613  
set\_mcmc\_num\_phases, 613  
set\_mcmc\_sigma\_scale, 613  
set\_param\_names, 614  
set\_prior, 614  
set\_rng\_seed, 614  
set\_ubound, 614

- sigma\_scale, 618
- theta\_in\_bounds, 614
- theta\_stack\_in\_bounds, 615
- ubound, 619
- mappel::MCMCAdaptor2Ds, 655
  - bound\_theta, 659
  - bounded\_theta, 659
  - bounded\_theta\_stack, 659
  - bounds\_epsilon, 669
  - check\_param\_shape, 660
  - check\_psf\_sigma, 660
  - default\_alpha\_sigma, 669
  - default\_beta\_pos, 670
  - default\_intensity\_kappa, 670
  - default\_max\_l, 670
  - default\_mean\_l, 670
  - default\_pixel\_mean\_bg, 670
  - default\_sigma\_pos, 670
  - DefaultSeperableInitEstimator, 671
  - eta\_bg, 671
  - eta\_l, 671
  - eta\_sigma, 671
  - eta\_x, 671
  - eta\_y, 672
  - get\_hyperparam\_index, 660
  - get\_hyperparam\_names, 660
  - get\_hyperparam\_value, 661
  - get\_hyperparams, 661
  - get\_lbound, 661
  - get\_mcmc\_num\_phases, 661
  - get\_mcmc\_sigma\_scale, 661
  - get\_num\_hyperparams, 661
  - get\_num\_params, 661
  - get\_param\_names, 662
  - get\_prior, 662
  - get\_rng\_generator, 662
  - get\_rng\_manager, 662
  - get\_stats, 662
  - get\_ubound, 662
  - global\_default\_mcmc\_sigma\_scale, 672
  - global\_max\_mcmc\_sigma\_scale, 672
  - global\_max\_psf\_sigma, 672
  - global\_min\_psf\_sigma, 672
  - has\_hyperparam, 663
  - lbound, 672
  - MCMCAdaptor2Ds, 659
  - make\_param, 663
  - make\_param\_mat, 663
  - make\_param\_mat\_stack, 663, 664
  - make\_param\_stack, 664
  - make\_prior\_component\_intensity, 664
  - make\_prior\_component\_position\_beta, 664
  - make\_prior\_component\_position\_normal, 665
  - make\_prior\_component\_sigma, 665
  - num\_hyperparams, 673
  - num\_params, 673
  - num\_phases, 673
  - operator=, 665
  - ParamVecT, 658
  - ParamT, 658
  - prior, 673
  - reflected\_theta, 665
  - reflected\_theta\_stack, 666
  - rename\_hyperparam, 666
  - sample\_mcmc\_candidate, 666
  - sample\_prior, 666
  - set\_background\_mcmc\_sampling, 667
  - set\_bounds, 667
  - set\_hyperparam\_names, 667
  - set\_hyperparam\_value, 667
  - set\_hyperparams, 667
  - set\_intensity\_mcmc\_sampling, 667
  - set\_lbound, 668
  - set\_mcmc\_num\_phases, 668
  - set\_mcmc\_sigma\_scale, 668
  - set\_param\_names, 668
  - set\_prior, 668
  - set\_rng\_seed, 669
  - set\_ubound, 669
  - sigma\_scale, 673
  - theta\_in\_bounds, 669
  - theta\_stack\_in\_bounds, 669
  - ubound, 674
- mappel::MCMCAdaptor2D, 637
  - bound\_theta, 641
  - bounded\_theta, 641
  - bounded\_theta\_stack, 641
  - bounds\_epsilon, 651
  - check\_param\_shape, 641
  - check\_psf\_sigma, 642
  - default\_alpha\_sigma, 651
  - default\_beta\_pos, 651
  - default\_intensity\_kappa, 651
  - default\_max\_l, 651
  - default\_mean\_l, 651
  - default\_pixel\_mean\_bg, 651
  - default\_sigma\_pos, 652
  - DefaultSeperableInitEstimator, 652
  - eta\_bg, 652
  - eta\_l, 652
  - eta\_x, 652
  - eta\_y, 653
  - get\_hyperparam\_index, 642
  - get\_hyperparam\_names, 642
  - get\_hyperparam\_value, 642
  - get\_hyperparams, 642
  - get\_lbound, 643
  - get\_mcmc\_num\_phases, 643

[get\\_mcmc\\_sigma\\_scale](#), 643  
[get\\_num\\_hyperparams](#), 643  
[get\\_num\\_params](#), 643  
[get\\_param\\_names](#), 643  
[get\\_prior](#), 643, 644  
[get\\_rng\\_generator](#), 644  
[get\\_rng\\_manager](#), 644  
[get\\_stats](#), 644  
[get\\_ubound](#), 644  
[global\\_default\\_mcmc\\_sigma\\_scale](#), 653  
[global\\_max\\_mcmc\\_sigma\\_scale](#), 653  
[global\\_max\\_psf\\_sigma](#), 653  
[global\\_min\\_psf\\_sigma](#), 653  
[has\\_hyperparam](#), 644  
[lbound](#), 653  
[MCMCAdaptor2D](#), 640, 641  
[make\\_param](#), 644, 645  
[make\\_param\\_mat](#), 645  
[make\\_param\\_mat\\_stack](#), 645  
[make\\_param\\_stack](#), 645  
[make\\_prior\\_component\\_intensity](#), 646  
[make\\_prior\\_component\\_position\\_beta](#), 646  
[make\\_prior\\_component\\_position\\_normal](#), 646  
[make\\_prior\\_component\\_sigma](#), 646  
[num\\_hyperparams](#), 654  
[num\\_params](#), 654  
[num\\_phases](#), 654  
[operator=](#), 646, 647  
[ParamVecT](#), 640  
[ParamT](#), 640  
[prior](#), 654  
[reflected\\_theta](#), 647  
[reflected\\_theta\\_stack](#), 647  
[rename\\_hyperparam](#), 647  
[sample\\_mcmc\\_candidate](#), 647  
[sample\\_prior](#), 648  
[set\\_background\\_mcmc\\_sampling](#), 648  
[set\\_bounds](#), 648  
[set\\_hyperparam\\_names](#), 648  
[set\\_hyperparam\\_value](#), 648  
[set\\_hyperparams](#), 649  
[set\\_intensity\\_mcmc\\_sampling](#), 649  
[set\\_lbound](#), 649  
[set\\_mcmc\\_num\\_phases](#), 649  
[set\\_mcmc\\_sigma\\_scale](#), 649  
[set\\_param\\_names](#), 649  
[set\\_prior](#), 650  
[set\\_rng\\_seed](#), 650  
[set\\_ubound](#), 650  
[sigma\\_scale](#), 654  
[theta\\_in\\_bounds](#), 650  
[theta\\_stack\\_in\\_bounds](#), 650  
[ubound](#), 655  
[mappel::MCMCAdaptorBase](#), 674  
[get\\_mcmc\\_num\\_phases](#), 675  
[get\\_mcmc\\_sigma\\_scale](#), 675  
[get\\_stats](#), 676  
[global\\_default\\_mcmc\\_sigma\\_scale](#), 676  
[global\\_max\\_mcmc\\_sigma\\_scale](#), 676  
[MCMCAdaptorBase](#), 675  
[num\\_phases](#), 676  
[set\\_mcmc\\_num\\_phases](#), 676  
[set\\_mcmc\\_sigma\\_scale](#), 676  
[sigma\\_scale](#), 677  
[mappel::MLEstimator](#), 683  
[bound\\_theta](#), 686  
[bounded\\_theta](#), 686  
[bounded\\_theta\\_stack](#), 687  
[bounds\\_epsilon](#), 695  
[check\\_param\\_shape](#), 687  
[check\\_psf\\_sigma](#), 687  
[default\\_alpha\\_sigma](#), 695  
[default\\_beta\\_pos](#), 695  
[default\\_intensity\\_kappa](#), 695  
[default\\_max\\_l](#), 695  
[default\\_mean\\_l](#), 696  
[default\\_pixel\\_mean\\_bg](#), 696  
[default\\_sigma\\_pos](#), 696  
[DefaultSeperableInitEstimator](#), 696  
[get\\_hyperparam\\_index](#), 687  
[get\\_hyperparam\\_names](#), 688  
[get\\_hyperparam\\_value](#), 688  
[get\\_hyperparams](#), 688  
[get\\_lbound](#), 688  
[get\\_num\\_hyperparams](#), 688  
[get\\_num\\_params](#), 688  
[get\\_param\\_names](#), 688  
[get\\_prior](#), 689  
[get\\_rng\\_generator](#), 689  
[get\\_rng\\_manager](#), 689  
[get\\_stats](#), 689  
[get\\_ubound](#), 689  
[global\\_max\\_psf\\_sigma](#), 696  
[global\\_min\\_psf\\_sigma](#), 696  
[has\\_hyperparam](#), 689  
[lbound](#), 697  
[MLEstimator](#), 686  
[make\\_param](#), 690  
[make\\_param\\_mat](#), 690  
[make\\_param\\_mat\\_stack](#), 690  
[make\\_param\\_stack](#), 690, 691  
[make\\_prior\\_component\\_intensity](#), 691  
[make\\_prior\\_component\\_position\\_beta](#), 691  
[make\\_prior\\_component\\_position\\_normal](#), 691  
[make\\_prior\\_component\\_sigma](#), 691  
[num\\_hyperparams](#), 697  
[num\\_params](#), 697  
[operator=](#), 692



- ParamVecT, 686
- ParamT, 686
- prior, 697
- reflected\_theta, 692
- reflected\_theta\_stack, 692
- rename\_hyperparam, 692
- sample\_prior, 692, 693
- set\_bounds, 693
- set\_hyperparam\_names, 693
- set\_hyperparam\_value, 693
- set\_hyperparams, 693
- set\_lbound, 693
- set\_param\_names, 694
- set\_prior, 694
- set\_rng\_seed, 694
- set\_ubound, 694
- theta\_in\_bounds, 694
- theta\_stack\_in\_bounds, 695
- ubound, 697
- mappel::ModelBoundsError, 698
  - ModelBoundsError, 698
- mappel::NotImplementedError, 741
  - NotImplementedError, 742
- mappel::NumericalError, 742
  - NumericalError, 742
- mappel::ParameterValueError, 745
  - ParameterValueError, 745
- mappel::PointEmitterModel, 746
  - bound\_theta, 749
  - bounded\_theta, 749
  - bounded\_theta\_stack, 750
  - bounds\_epsilon, 758
  - check\_param\_shape, 750
  - check\_psf\_sigma, 750
  - default\_alpha\_sigma, 758
  - default\_beta\_pos, 758
  - default\_intensity\_kappa, 758
  - default\_max\_l, 758
  - default\_mean\_l, 758
  - default\_pixel\_mean\_bg, 758
  - default\_sigma\_pos, 759
  - DefaultSeperableInitEstimator, 759
  - get\_hyperparam\_index, 750
  - get\_hyperparam\_names, 751
  - get\_hyperparam\_value, 751
  - get\_hyperparams, 751
  - get\_lbound, 751
  - get\_num\_hyperparams, 751
  - get\_num\_params, 751
  - get\_param\_names, 751
  - get\_prior, 752
  - get\_rng\_generator, 752
  - get\_rng\_manager, 752
  - get\_stats, 752
  - get\_ubound, 752
  - global\_max\_psf\_sigma, 759
  - global\_min\_psf\_sigma, 759
  - has\_hyperparam, 752
  - lbound, 759
  - make\_param, 753
  - make\_param\_mat, 753
  - make\_param\_mat\_stack, 753
  - make\_param\_stack, 753, 754
  - make\_prior\_component\_intensity, 754
  - make\_prior\_component\_position\_beta, 754
  - make\_prior\_component\_position\_normal, 754
  - make\_prior\_component\_sigma, 754
  - num\_hyperparams, 759
  - num\_params, 760
  - operator=, 755
  - ParamVecT, 749
  - ParamT, 749
  - PointEmitterModel, 749
  - prior, 760
  - reflected\_theta, 755
  - reflected\_theta\_stack, 755
  - rename\_hyperparam, 755
  - sample\_prior, 755, 756
  - set\_bounds, 756
  - set\_hyperparam\_names, 756
  - set\_hyperparam\_value, 756
  - set\_hyperparams, 756
  - set\_lbound, 756
  - set\_param\_names, 756
  - set\_prior, 757
  - set\_rng\_seed, 757
  - set\_ubound, 757
  - theta\_in\_bounds, 757
  - theta\_stack\_in\_bounds, 757
  - ubound, 760
- mappel::PoissonNoise1DObjective, 763
  - check\_image\_shape, 766
  - check\_size, 767
  - estimator\_names, 769
  - get\_image\_from\_stack, 767
  - get\_num\_pixels, 767
  - get\_size, 767
  - get\_size\_image\_stack, 767
  - get\_stats, 767
  - global\_max\_size, 769
  - global\_min\_size, 769
  - ImageCoordT, 764
  - ImagePixelT, 764
  - ImageShapeT, 764
  - ImageSizeShapeT, 765
  - ImageSizeVecShapeT, 765
  - ImageSizeVecT, 765
  - ImageSizeT, 765

- ImageStackShapeT, 765
- ImageStackT, 765
- ImageT, 765
- make\_image, 768
- make\_image\_stack, 768
- ModelDataStackT, 766
- ModelDataT, 766
- num\_dim, 769
- operator=, 768
- PoissonNoise1DObjective, 766
- set\_image\_in\_stack, 768
- set\_size, 768
- size, 769
- mappel::PoissonNoise2DObjective, 770
  - check\_image\_shape, 773
  - check\_size, 774
  - estimator\_names, 776
  - get\_image\_from\_stack, 774
  - get\_num\_pixels, 774
  - get\_size, 774
  - get\_size\_image\_stack, 774
  - get\_stats, 774
  - global\_max\_size, 776
  - global\_min\_size, 776
  - ImageCoordT, 771
  - ImagePixelT, 771
  - ImageShapeT, 771
  - ImageSizeShapeT, 772
  - ImageSizeVecShapeT, 772
  - ImageSizeVecT, 772
  - ImageSizeT, 772
  - ImageStackShapeT, 772
  - ImageStackT, 772
  - ImageT, 772
  - make\_image, 775
  - make\_image\_stack, 775
  - ModelDataStackT, 773
  - ModelDataT, 773
  - num\_dim, 776
  - operator=, 775
  - PoissonNoise2DObjective, 773
  - set\_image\_in\_stack, 775
  - set\_size, 775
  - size, 776
- mappel::estimator, 47
  - Error, 50
  - ExitCode, 50
  - FunctionValue, 50
  - GradRatio, 50
  - MaxBacktracks, 50
  - MaxIter, 50
  - ModelImprovement, 50
  - operator<, 50
  - StepSize, 50
  - Success, 50
  - TrustRegionRadius, 50
  - Unassigned, 50
- mappel::estimator::CGaussHeuristicEstimator
  - CGaussHeuristicEstimator, 94
  - clear\_stats, 94
  - compute\_estimate\_debug, 94
  - compute\_profile\_bound, 95
  - compute\_profile\_bound\_debug, 95
  - compute\_profile\_estimate, 95
  - estimate\_max, 95, 96
  - estimate\_max\_debug, 97
  - estimate\_max\_stack, 98
  - estimate\_profile\_bounds, 99
  - estimate\_profile\_bounds\_debug, 99
  - estimate\_profile\_bounds\_parallel, 100
  - estimate\_profile\_bounds\_stack, 100
  - estimate\_profile\_max, 101
  - exit\_counts, 103
  - get\_debug\_stats, 101
  - get\_exit\_counts, 101
  - get\_model, 102
  - get\_stats, 102
  - max\_threads, 103
  - model, 103
  - mtx, 103
  - name, 102
  - num\_estimations, 104
  - num\_threads, 104
  - record\_exit\_code, 102
  - record\_walltime, 102
  - total\_walltime, 104
- mappel::estimator::CGaussHeuristicEstimator< Model >, 93
- mappel::estimator::CGaussMLE< Model >, 105
- mappel::estimator::CGaussMLE
  - CGaussMLE, 106
  - clear\_stats, 107
  - compute\_profile\_bound, 107
  - compute\_profile\_bound\_debug, 107
  - compute\_profile\_estimate, 107
  - DefaultIterations, 115
  - estimate\_max, 107, 108
  - estimate\_max\_debug, 109
  - estimate\_max\_stack, 110
  - estimate\_profile\_bounds, 111
  - estimate\_profile\_bounds\_debug, 111
  - estimate\_profile\_bounds\_parallel, 112
  - estimate\_profile\_bounds\_stack, 112
  - estimate\_profile\_max, 113
  - exit\_counts, 115
  - get\_debug\_stats, 113
  - get\_exit\_counts, 114
  - get\_model, 114

- get\_stats, 114
- max\_threads, 115
- model, 116
- mtx, 116
- name, 114
- num\_estimations, 116
- num\_threads, 116
- record\_exit\_code, 114
- record\_walltime, 115
- total\_walltime, 117
- mappel::estimator::Estimator
  - ~Estimator, 119
  - clear\_stats, 119
  - compute\_estimate, 119
  - compute\_estimate\_debug, 120
  - compute\_profile\_bound, 120
  - compute\_profile\_bound\_debug, 120
  - compute\_profile\_estimate, 120
  - estimate\_max, 121, 122
  - estimate\_max\_debug, 122, 123
  - estimate\_max\_stack, 123, 124
  - estimate\_profile\_bounds, 124
  - estimate\_profile\_bounds\_debug, 124
  - estimate\_profile\_bounds\_parallel, 125
  - estimate\_profile\_bounds\_stack, 125
  - estimate\_profile\_max, 125
  - Estimator, 119
  - exit\_counts, 127
  - get\_debug\_stats, 125
  - get\_exit\_counts, 126
  - get\_model, 126
  - get\_stats, 126
  - model, 127
  - name, 126
  - num\_estimations, 128
  - operator<<, 127
  - record\_exit\_code, 126
  - record\_walltime, 127
  - total\_walltime, 128
- mappel::estimator::Estimator< Model >, 117
- mappel::estimator::HeuristicEstimator
  - clear\_stats, 532
  - compute\_estimate\_debug, 532
  - compute\_profile\_bound, 533
  - compute\_profile\_bound\_debug, 533
  - compute\_profile\_estimate, 533
  - estimate\_max, 533, 534
  - estimate\_max\_debug, 535
  - estimate\_max\_stack, 536
  - estimate\_profile\_bounds, 537
  - estimate\_profile\_bounds\_debug, 537
  - estimate\_profile\_bounds\_parallel, 538
  - estimate\_profile\_bounds\_stack, 538
  - estimate\_profile\_max, 539
  - exit\_counts, 541
  - get\_debug\_stats, 539
  - get\_exit\_counts, 540
  - get\_model, 540
  - get\_stats, 540
  - HeuristicEstimator, 532
  - max\_threads, 541
  - model, 541
  - mtx, 542
  - name, 540
  - num\_estimations, 542
  - num\_threads, 542
  - record\_exit\_code, 540
  - record\_walltime, 541
  - total\_walltime, 542
- mappel::estimator::HeuristicEstimator< Model >, 530
- mappel::estimator::IterativeMaximizer
  - backtrack, 559
  - backtrack\_max\_ratio, 572
  - backtrack\_min\_linear\_step\_ratio, 572
  - backtrack\_min\_ratio, 573
  - clear\_stats, 559
  - compute\_estimate, 560
  - compute\_estimate\_debug, 560
  - compute\_profile\_bound, 560
  - compute\_profile\_bound\_debug, 561
  - compute\_profile\_estimate, 561
  - convergence\_min\_function\_change\_ratio, 573
  - convergence\_min\_step\_size\_ratio, 573
  - convergence\_test\_grad\_ratio, 561
  - convergence\_test\_step\_size, 562
  - DefaultIterations, 573
  - estimate\_max, 562, 563
  - estimate\_max\_debug, 563, 564
  - estimate\_max\_stack, 564, 565
  - estimate\_profile\_bounds, 565
  - estimate\_profile\_bounds\_debug, 566
  - estimate\_profile\_bounds\_parallel, 566
  - estimate\_profile\_bounds\_stack, 566
  - estimate\_profile\_max, 567
  - exit\_counts, 573
  - get\_debug\_stats, 567
  - get\_exit\_counts, 568
  - get\_model, 568
  - get\_stats, 568
  - get\_total\_backtracks, 568
  - get\_total\_der\_evals, 569
  - get\_total\_fun\_evals, 569
  - get\_total\_iterations, 569
  - IterativeMaximizer, 559
  - last\_backtrack\_idx, 573
  - local\_maximize, 569
  - local\_profile\_maximize, 569
  - max\_backtracks, 574

- max\_iterations, 574
- max\_threads, 574
- maximize, 570
- mean\_backtracks, 570
- mean\_der\_evals, 571
- mean\_fun\_evals, 571
- mean\_iterations, 571
- min\_eigenvalue\_correction\_delta, 574
- min\_profile\_bound\_residual, 574
- model, 574
- mtx, 575
- name, 571
- num\_estimations, 575
- num\_threads, 575
- profile\_bound\_backtrack, 571
- record\_exit\_code, 571
- record\_run\_statistics, 571
- record\_walltime, 572
- solve\_profile\_bound, 572
- total\_backtracks, 575
- total\_der\_evals, 576
- total\_fun\_evals, 576
- total\_iterations, 576
- total\_walltime, 576
- mappel::estimator::IterativeMaximizer< Model >, 556
- mappel::estimator::IterativeMaximizer< Model >::←  
MaximizerData, 592
- mappel::estimator::IterativeMaximizer::MaximizerData
  - backtrack\_idx, 598
  - current\_stencil, 598
  - DefaultMaxSeqLength, 598
  - expand\_max\_seq\_len, 594
  - fixed\_idx, 598
  - free\_idx, 599
  - get\_backtrack\_idx, 594
  - get\_sequence\_len, 594
  - get\_theta\_sequence, 594
  - get\_theta\_sequence\_rllh, 595
  - grad, 599
  - has\_fixed\_parameters, 595
  - has\_theta\_sequence, 595
  - im, 599
  - max\_seq\_len, 599
  - MaximizerData, 594
  - nBacktracks, 599
  - nIterations, 599
  - num\_fixed\_parameters, 595
  - num\_params, 600
  - record\_backtrack, 595
  - record\_iteration, 596
  - restore\_stencil, 596
  - rllh, 600
  - s0, 600
  - s1, 600
  - save\_stencil, 596
  - saved\_stencil, 597
  - saved\_theta, 597
  - seq\_len, 600
  - seq\_rllh, 600
  - set\_fixed\_parameters, 597
  - set\_stencil, 597
  - stencil, 597
  - step, 601
  - theta, 598
  - theta\_seq, 601
- mappel::estimator::MLEData, 48, 880
- mappel::estimator::MLEDataStack, 48, 881
- mappel::estimator::MLEDebugData, 48, 880
- mappel::estimator::NewtonDiagonalMaximizer
  - backtrack, 702
  - backtrack\_max\_ratio, 716
  - backtrack\_min\_linear\_step\_ratio, 716
  - backtrack\_min\_ratio, 716
  - clear\_stats, 702
  - compute\_estimate, 702
  - compute\_estimate\_debug, 703
  - compute\_profile\_bound, 703
  - compute\_profile\_bound\_debug, 703
  - compute\_profile\_estimate, 704
  - convergence\_min\_function\_change\_ratio, 716
  - convergence\_min\_step\_size\_ratio, 716
  - convergence\_test\_grad\_ratio, 704
  - convergence\_test\_step\_size, 704
  - DefaultIterations, 717
  - estimate\_max, 704, 705
  - estimate\_max\_debug, 706
  - estimate\_max\_stack, 707
  - estimate\_profile\_bounds, 708
  - estimate\_profile\_bounds\_debug, 708
  - estimate\_profile\_bounds\_parallel, 709
  - estimate\_profile\_bounds\_stack, 709
  - estimate\_profile\_max, 710
  - exit\_counts, 717
  - get\_debug\_stats, 710
  - get\_exit\_counts, 711
  - get\_model, 711
  - get\_stats, 711
  - get\_total\_backtracks, 711
  - get\_total\_der\_evals, 712
  - get\_total\_fun\_evals, 712
  - get\_total\_iterations, 712
  - last\_backtrack\_idx, 717
  - local\_maximize, 712
  - local\_profile\_maximize, 712
  - max\_backtracks, 717
  - max\_iterations, 717
  - max\_threads, 717
  - maximize, 713

- MaximizerData, [701](#)
- mean\_backtracks, [713](#)
- mean\_der\_evals, [714](#)
- mean\_fun\_evals, [714](#)
- mean\_iterations, [714](#)
- min\_eigenvalue\_correction\_delta, [718](#)
- min\_profile\_bound\_residual, [718](#)
- model, [718](#)
- mtx, [718](#)
- name, [714](#)
- NewtonDiagonalMaximizer, [702](#)
- num\_estimations, [718](#)
- num\_threads, [719](#)
- profile\_bound\_backtrack, [714](#)
- record\_exit\_code, [714](#)
- record\_run\_statistics, [715](#)
- record\_walltime, [715](#)
- solve\_profile\_bound, [715](#)
- total\_backtracks, [719](#)
- total\_der\_evals, [719](#)
- total\_fun\_evals, [719](#)
- total\_iterations, [719](#)
- total\_walltime, [720](#)
- mappel::estimator::NewtonDiagonalMaximizer< Model >, [699](#)
- mappel::estimator::NewtonMaximizer
  - backtrack, [724](#)
  - backtrack\_max\_ratio, [737](#)
  - backtrack\_min\_linear\_step\_ratio, [737](#)
  - backtrack\_min\_ratio, [737](#)
  - clear\_stats, [724](#)
  - compute\_estimate, [724](#)
  - compute\_estimate\_debug, [724](#)
  - compute\_profile\_bound, [725](#)
  - compute\_profile\_bound\_debug, [725](#)
  - compute\_profile\_estimate, [725](#)
  - convergence\_min\_function\_change\_ratio, [737](#)
  - convergence\_min\_step\_size\_ratio, [737](#)
  - convergence\_test\_grad\_ratio, [726](#)
  - convergence\_test\_step\_size, [726](#)
  - DefaultIterations, [738](#)
  - estimate\_max, [726](#), [727](#)
  - estimate\_max\_debug, [728](#)
  - estimate\_max\_stack, [729](#)
  - estimate\_profile\_bounds, [730](#)
  - estimate\_profile\_bounds\_debug, [730](#)
  - estimate\_profile\_bounds\_parallel, [730](#)
  - estimate\_profile\_bounds\_stack, [730](#)
  - estimate\_profile\_max, [731](#)
  - exit\_counts, [738](#)
  - get\_debug\_stats, [731](#)
  - get\_exit\_counts, [732](#)
  - get\_model, [732](#)
  - get\_stats, [732](#)
  - get\_total\_backtracks, [732](#)
  - get\_total\_der\_evals, [733](#)
  - get\_total\_fun\_evals, [733](#)
  - get\_total\_iterations, [733](#)
  - last\_backtrack\_idx, [738](#)
  - local\_maximize, [733](#)
  - local\_profile\_maximize, [733](#)
  - max\_backtracks, [738](#)
  - max\_iterations, [738](#)
  - max\_threads, [738](#)
  - maximize, [734](#)
  - MaximizerData, [723](#)
  - mean\_backtracks, [734](#)
  - mean\_der\_evals, [735](#)
  - mean\_fun\_evals, [735](#)
  - mean\_iterations, [735](#)
  - min\_eigenvalue\_correction\_delta, [739](#)
  - min\_profile\_bound\_residual, [739](#)
  - model, [739](#)
  - mtx, [739](#)
  - name, [735](#)
  - NewtonMaximizer, [723](#)
  - num\_estimations, [739](#)
  - num\_threads, [740](#)
  - profile\_bound\_backtrack, [735](#)
  - record\_exit\_code, [735](#)
  - record\_run\_statistics, [736](#)
  - record\_walltime, [736](#)
  - solve\_profile\_bound, [736](#)
  - total\_backtracks, [740](#)
  - total\_der\_evals, [740](#)
  - total\_fun\_evals, [740](#)
  - total\_iterations, [740](#)
  - total\_walltime, [741](#)
- mappel::estimator::NewtonMaximizer< Model >, [720](#)
- mappel::estimator::ProfileBoundsData, [777](#)
  - confidence, [778](#)
  - estimated\_idx, [778](#)
  - initialize\_arrays, [778](#)
  - mle, [778](#)
  - Nparams\_est, [778](#)
  - profile\_lb, [779](#)
  - profile\_points\_lb, [779](#)
  - profile\_points\_lb\_rllh, [779](#)
  - profile\_points\_ub, [779](#)
  - profile\_points\_ub\_rllh, [779](#)
  - profile\_ub, [780](#)
  - target\_rllh\_delta, [780](#)
- mappel::estimator::ProfileBoundsDataStack, [780](#)
  - confidence, [782](#)
  - estimated\_idx, [782](#)
  - initialize\_arrays, [781](#)
  - mle, [782](#)
  - Ndata, [782](#)

- Nparams\_est, 782
- profile\_lb, 782
- profile\_points\_lb, 782
- profile\_points\_lb\_rllh, 783
- profile\_points\_ub, 783
- profile\_points\_ub\_rllh, 783
- profile\_ub, 783
- target\_rllh\_delta, 783
- mappel::estimator::ProfileBoundsDebugData, 49, 881
- mappel::estimator::ProfileLikelihoodData, 49, 881
- mappel::estimator::QuasiNewtonMaximizer
  - backtrack, 787
  - backtrack\_max\_ratio, 801
  - backtrack\_min\_linear\_step\_ratio, 801
  - backtrack\_min\_ratio, 801
  - clear\_stats, 787
  - compute\_estimate, 787
  - compute\_estimate\_debug, 788
  - compute\_profile\_bound, 788
  - compute\_profile\_bound\_debug, 788
  - compute\_profile\_estimate, 789
  - convergence\_min\_function\_change\_ratio, 801
  - convergence\_min\_step\_size\_ratio, 801
  - convergence\_test\_grad\_ratio, 789
  - convergence\_test\_step\_size, 789
  - DefaultIterations, 802
  - estimate\_max, 789, 790
  - estimate\_max\_debug, 791
  - estimate\_max\_stack, 792
  - estimate\_profile\_bounds, 793
  - estimate\_profile\_bounds\_debug, 793
  - estimate\_profile\_bounds\_parallel, 794
  - estimate\_profile\_bounds\_stack, 794
  - estimate\_profile\_max, 795
  - exit\_counts, 802
  - get\_debug\_stats, 795
  - get\_exit\_counts, 796
  - get\_model, 796
  - get\_stats, 796
  - get\_total\_backtracks, 796
  - get\_total\_der\_evals, 797
  - get\_total\_fun\_evals, 797
  - get\_total\_iterations, 797
  - last\_backtrack\_idx, 802
  - local\_maximize, 797
  - local\_profile\_maximize, 797
  - max\_backtracks, 802
  - max\_iterations, 802
  - max\_threads, 802
  - maximize, 798
  - MaximizerData, 786
  - mean\_backtracks, 798
  - mean\_der\_evals, 799
  - mean\_fun\_evals, 799
  - mean\_iterations, 799
  - min\_eigenvalue\_correction\_delta, 803
  - min\_profile\_bound\_residual, 803
  - model, 803
  - mtx, 803
  - name, 799
  - num\_estimations, 803
  - num\_threads, 804
  - profile\_bound\_backtrack, 799
  - QuasiNewtonMaximizer, 787
  - record\_exit\_code, 799
  - record\_run\_statistics, 800
  - record\_walltime, 800
  - solve\_profile\_bound, 800
  - total\_backtracks, 804
  - total\_der\_evals, 804
  - total\_fun\_evals, 804
  - total\_iterations, 804
  - total\_walltime, 805
- mappel::estimator::QuasiNewtonMaximizer< Model >, 784
- mappel::estimator::SimulatedAnnealingMaximizer
  - clear\_stats, 807
  - compute\_profile\_bound, 807
  - compute\_profile\_bound\_debug, 808
  - Default\_T\_Init, 815
  - DefaultCoolingRate, 815
  - DefaultNumIterations, 815
  - estimate\_max, 808, 809
  - estimate\_max\_debug, 809, 810
  - estimate\_max\_stack, 810, 811
  - estimate\_profile\_bounds, 811
  - estimate\_profile\_bounds\_debug, 812
  - estimate\_profile\_bounds\_parallel, 812
  - estimate\_profile\_bounds\_stack, 812
  - estimate\_profile\_max, 813
  - exit\_counts, 816
  - get\_debug\_stats, 813
  - get\_exit\_counts, 814
  - get\_model, 814
  - get\_stats, 814
  - max\_threads, 816
  - model, 816
  - mtx, 816
  - name, 814
  - num\_estimations, 816
  - num\_threads, 817
  - record\_exit\_code, 814
  - record\_walltime, 815
  - SimulatedAnnealingMaximizer, 807
  - total\_walltime, 817
- mappel::estimator::SimulatedAnnealingMaximizer< Model >, 805
- mappel::estimator::ThreadedEstimator

- clear\_stats, 842
- compute\_estimate, 842
- compute\_estimate\_debug, 843
- compute\_profile\_bound, 843
- compute\_profile\_bound\_debug, 843
- compute\_profile\_estimate, 843
- estimate\_max, 844, 845
- estimate\_max\_debug, 845, 846
- estimate\_max\_stack, 846, 847
- estimate\_profile\_bounds, 847
- estimate\_profile\_bounds\_debug, 847
- estimate\_profile\_bounds\_parallel, 848
- estimate\_profile\_bounds\_stack, 848
- estimate\_profile\_max, 849
- exit\_counts, 851
- get\_debug\_stats, 849
- get\_exit\_counts, 850
- get\_model, 850
- get\_stats, 850
- max\_threads, 851
- model, 852
- mtx, 852
- name, 850
- num\_estimations, 852
- num\_threads, 852
- record\_exit\_code, 851
- record\_walltime, 851
- ThreadedEstimator, 842
- total\_walltime, 853
- mappel::estimator::ThreadedEstimator< Model >, 840
- mappel::estimator::TrustRegionMaximizer
  - backtrack, 857
  - backtrack\_max\_ratio, 871
  - backtrack\_min\_linear\_step\_ratio, 871
  - backtrack\_min\_ratio, 871
  - clear\_stats, 857
  - compute\_estimate, 857
  - compute\_estimate\_debug, 858
  - compute\_profile\_bound, 858
  - compute\_profile\_bound\_debug, 858
  - compute\_profile\_estimate, 859
  - convergence\_min\_function\_change\_ratio, 871
  - convergence\_min\_step\_size\_ratio, 871
  - convergence\_min\_trust\_radius, 872
  - convergence\_test\_grad\_ratio, 859
  - convergence\_test\_step\_size, 859
  - DefaultIterations, 872
  - estimate\_max, 859, 860
  - estimate\_max\_debug, 861
  - estimate\_max\_stack, 862
  - estimate\_profile\_bounds, 863
  - estimate\_profile\_bounds\_debug, 863
  - estimate\_profile\_bounds\_parallel, 864
  - estimate\_profile\_bounds\_stack, 864
  - estimate\_profile\_max, 865
  - exit\_counts, 872
  - get\_debug\_stats, 865
  - get\_exit\_counts, 866
  - get\_model, 866
  - get\_stats, 866
  - get\_total\_backtracks, 866
  - get\_total\_der\_evals, 867
  - get\_total\_fun\_evals, 867
  - get\_total\_iterations, 867
  - last\_backtrack\_idx, 872
  - local\_maximize, 867
  - local\_profile\_maximize, 867
  - max\_backtracks, 872
  - max\_iterations, 872
  - max\_threads, 873
  - maximize, 868
  - MaximizerData, 856
  - mean\_backtracks, 868
  - mean\_der\_evals, 869
  - mean\_fun\_evals, 869
  - mean\_iterations, 869
  - min\_eigenvalue\_correction\_delta, 873
  - min\_profile\_bound\_residual, 873
  - model, 873
  - mtx, 873
  - name, 869
  - num\_estimations, 874
  - num\_threads, 874
  - profile\_bound\_backtrack, 869
  - record\_exit\_code, 869
  - record\_run\_statistics, 870
  - record\_walltime, 870
  - rho\_cauchy\_min, 874
  - rho\_obj\_min, 874
  - rho\_obj\_opt, 874
  - solve\_profile\_bound, 870
  - total\_backtracks, 875
  - total\_der\_evals, 875
  - total\_fun\_evals, 875
  - total\_iterations, 875
  - total\_walltime, 875
  - trust\_radius\_decrease, 875
  - trust\_radius\_decrease\_min, 876
  - trust\_radius\_increase, 876
  - TrustRegionMaximizer, 857
- mappel::estimator::TrustRegionMaximizer< Model >, 853
- mappel::estimator::subroutine, 51
  - bound\_step, 51
  - compute\_D\_scale, 52
  - compute\_bound\_scaling\_vec, 52
  - compute\_cauchy\_point, 52
  - compute\_initial\_trust\_radius, 52
  - compute\_quadratic\_model\_value, 53



- compute\_scaled\_problem, 53
- solve\_TR\_subproblem, 53
- solve\_profile\_initial\_step, 53
- solve\_restricted\_step\_length\_newton, 53
- mappel::mcmc, 54
  - compute\_posterior\_credible, 54
  - estimate\_sample\_posterior, 54
  - num\_oversample, 55
  - sample\_posterior, 55
  - sample\_posterior\_debug, 55
  - thin\_sample, 55
- mappel::mcmc::MCMCData, 677
  - confidence, 678
  - credible\_lb, 678
  - credible\_ub, 678
  - initialize\_arrays, 678
  - Nburnin, 678
  - Nsample, 678
  - sample, 678
  - sample\_cov, 679
  - sample\_mean, 679
  - sample\_rllh, 679
  - thin, 679
- mappel::mcmc::MCMCDataStack, 679
  - confidence, 680
  - credible\_lb, 680
  - credible\_ub, 680
  - initialize\_arrays, 680
  - Nburnin, 680
  - Ndata, 681
  - Nsample, 681
  - sample, 681
  - sample\_cov, 681
  - sample\_mean, 681
  - sample\_rllh, 681
  - thin, 681
- mappel::mcmc::MCMCDebugData, 682
  - candidate, 682
  - candidate\_rllh, 682
  - initialize\_arrays, 682
  - Nsample, 683
  - sample, 683
  - sample\_rllh, 683
- mappel::methods, 55
  - aposteriori\_objective, 59
  - cr\_lower\_bound, 59
  - error\_bounds\_expected, 59
  - error\_bounds\_observed, 59
  - error\_bounds\_posterior\_credible, 60
  - error\_bounds\_profile\_likelihood, 60
  - estimate\_max, 60, 61
  - estimate\_posterior, 61
  - estimate\_profile\_likelihood, 61, 62
  - expected\_information, 62
  - likelihood\_objective, 63
  - make\_estimator, 63
  - model\_image, 63, 64
  - observed\_information, 64
  - prior\_objective, 64
  - simulate\_image, 64–66
  - simulate\_image\_from\_model, 66
- mappel::methods::debug, 67
  - error\_bounds\_profile\_likelihood\_debug, 67
  - estimate\_max\_debug, 67
  - estimate\_posterior\_debug, 68
- mappel::methods::likelihood, 68
  - grad, 69
  - grad2, 69
  - hessian, 69
  - llh, 70
  - rllh, 70
- mappel::methods::likelihood::debug, 71
  - grad\_components, 71
  - hessian\_components, 71
  - llh\_components, 72
  - rllh\_components, 72
- mappel::methods::objective, 72
  - grad, 74
  - grad2, 74, 75
  - hessian, 75, 76
  - llh, 76
  - negative\_definite\_hessian, 76, 77
  - rllh, 77
- mappel::methods::objective::debug, 78
  - grad\_components, 78, 79
  - hessian\_components, 79
  - llh\_components, 79
  - rllh\_components, 80
- mappel::methods::objective::openmp, 80
  - grad\_stack, 81
  - hessian\_stack, 81
  - llh\_stack, 82
  - negative\_definite\_hessian\_stack, 83
  - rllh\_stack, 83, 84
- mappel::methods::openmp, 84
  - cr\_lower\_bound\_stack, 85
  - error\_bounds\_expected\_stack, 86
  - error\_bounds\_observed\_stack, 86
  - error\_bounds\_profile\_likelihood\_parallel, 86
  - error\_bounds\_profile\_likelihood\_stack, 86, 87
  - estimate\_max\_stack, 87
  - estimate\_posterior\_stack, 87, 88
  - estimate\_profile\_likelihood\_stack, 88
  - expected\_information\_stack, 88
  - model\_image\_stack, 88
  - sample\_prior\_stack, 89
  - simulate\_image\_stack, 89
- MappelError



- mappel, [27](#)
- MatT
  - mappel, [28](#)
- max\_backtracks
  - mappel::estimator::IterativeMaximizer, [574](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [717](#)
  - mappel::estimator::NewtonMaximizer, [738](#)
  - mappel::estimator::QuasiNewtonMaximizer, [802](#)
  - mappel::estimator::TrustRegionMaximizer, [872](#)
- max\_iterations
  - mappel::estimator::IterativeMaximizer, [574](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [717](#)
  - mappel::estimator::NewtonMaximizer, [738](#)
  - mappel::estimator::QuasiNewtonMaximizer, [802](#)
  - mappel::estimator::TrustRegionMaximizer, [872](#)
- max\_seq\_len
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, [599](#)
- max\_threads
  - mappel::estimator::CGaussHeuristicEstimator, [103](#)
  - mappel::estimator::CGaussMLE, [115](#)
  - mappel::estimator::HeuristicEstimator, [541](#)
  - mappel::estimator::IterativeMaximizer, [574](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [717](#)
  - mappel::estimator::NewtonMaximizer, [738](#)
  - mappel::estimator::QuasiNewtonMaximizer, [802](#)
  - mappel::estimator::SimulatedAnnealingMaximizer, [816](#)
  - mappel::estimator::ThreadedEstimator, [851](#)
  - mappel::estimator::TrustRegionMaximizer, [873](#)
- MaxBacktracks
  - mappel::estimator, [50](#)
- MaxIter
  - mappel::estimator, [50](#)
- maxidx
  - mappel, [40](#)
- maximize
  - mappel::estimator::IterativeMaximizer, [570](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [713](#)
  - mappel::estimator::NewtonMaximizer, [734](#)
  - mappel::estimator::QuasiNewtonMaximizer, [798](#)
  - mappel::estimator::TrustRegionMaximizer, [868](#)
- MaximizerData
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, [594](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [701](#)
  - mappel::estimator::NewtonMaximizer, [723](#)
  - mappel::estimator::QuasiNewtonMaximizer, [786](#)
  - mappel::estimator::TrustRegionMaximizer, [856](#)
- mcmc.cpp, [906](#)
- mcmc.h, [907](#)
- mcmc\_candidate\_eta\_sigma
  - mappel::Gauss2DsxyModel, [528](#)
- mcmc\_candidate\_eta\_y
  - mappel::Gauss2DsxyModel, [528](#)
- mcmc\_data.h, [908](#)
- mean\_backtracks
  - mappel::estimator::IterativeMaximizer, [570](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [713](#)
  - mappel::estimator::NewtonMaximizer, [734](#)
  - mappel::estimator::QuasiNewtonMaximizer, [798](#)
  - mappel::estimator::TrustRegionMaximizer, [868](#)
- mean\_der\_evals
  - mappel::estimator::IterativeMaximizer, [571](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [714](#)
  - mappel::estimator::NewtonMaximizer, [735](#)
  - mappel::estimator::QuasiNewtonMaximizer, [799](#)
  - mappel::estimator::TrustRegionMaximizer, [869](#)
- mean\_fun\_evals
  - mappel::estimator::IterativeMaximizer, [571](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [714](#)
  - mappel::estimator::NewtonMaximizer, [735](#)
  - mappel::estimator::QuasiNewtonMaximizer, [799](#)
  - mappel::estimator::TrustRegionMaximizer, [869](#)
- mean\_iterations
  - mappel::estimator::IterativeMaximizer, [571](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [714](#)
  - mappel::estimator::NewtonMaximizer, [735](#)
  - mappel::estimator::QuasiNewtonMaximizer, [799](#)
  - mappel::estimator::TrustRegionMaximizer, [869](#)
- min\_eigenvalue\_correction\_delta
  - mappel::estimator::IterativeMaximizer, [574](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [718](#)
  - mappel::estimator::NewtonMaximizer, [739](#)
  - mappel::estimator::QuasiNewtonMaximizer, [803](#)
  - mappel::estimator::TrustRegionMaximizer, [873](#)
- min\_profile\_bound\_residual
  - mappel::estimator::IterativeMaximizer, [574](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [718](#)
  - mappel::estimator::NewtonMaximizer, [739](#)
  - mappel::estimator::QuasiNewtonMaximizer, [803](#)
  - mappel::estimator::TrustRegionMaximizer, [873](#)
- min\_sigma
  - mappel::Gauss2DsMAP, [413](#)
  - mappel::Gauss2DsMLE, [446](#)
  - mappel::Gauss2DsModel, [477](#)
  - mappel::Gauss2DsxyModel, [528](#)
- mle
  - mappel::estimator::ProfileBoundsData, [778](#)
  - mappel::estimator::ProfileBoundsDataStack, [782](#)
- model
  - mappel::Gauss1DModel::Stencil, [840](#)
  - mappel::Gauss1DsModel::Stencil, [826](#)
  - mappel::Gauss2DModel::Stencil, [830](#)
  - mappel::Gauss2DsModel::Stencil, [836](#)
  - mappel::Gauss2DsxyModel::Stencil, [821](#)
  - mappel::estimator::CGaussHeuristicEstimator, [103](#)
  - mappel::estimator::CGaussMLE, [116](#)

- mappel::estimator::Estimator, 127
- mappel::estimator::HeuristicEstimator, 541
- mappel::estimator::IterativeMaximizer, 574
- mappel::estimator::NewtonDiagonalMaximizer, 718
- mappel::estimator::NewtonMaximizer, 739
- mappel::estimator::QuasiNewtonMaximizer, 803
- mappel::estimator::SimulatedAnnealingMaximizer, 816
- mappel::estimator::ThreadedEstimator, 852
- mappel::estimator::TrustRegionMaximizer, 873
- model\_grad
  - PoissonGaussianNoise2DObjective.h, 931
- model\_grad2
  - PoissonGaussianNoise2DObjective.h, 931
- model\_hessian
  - PoissonGaussianNoise2DObjective.h, 931
- model\_image
  - mappel::methods, 63, 64
  - PoissonGaussianNoise2DObjective.h, 931
- model\_image\_stack
  - mappel::methods::openmp, 88
- model\_methods.h, 915
- model\_methods\_impl.h, 918
- ModelBoundsError
  - mappel::ModelBoundsError, 698
- ModelDataStackT
  - mappel, 28
  - mappel::Gauss1DMAP, 134
  - mappel::Gauss1DMLE, 162
  - mappel::Gauss1DsMAP, 216
  - mappel::Gauss1DsMLE, 245
  - mappel::Gauss2DMAP, 301
  - mappel::Gauss2DMLE, 331
  - mappel::Gauss2DsMAP, 391
  - mappel::Gauss2DsMLE, 423
  - mappel::Gauss2DsxyMAP, 485
  - mappel::PoissonNoise1DObjective, 766
  - mappel::PoissonNoise2DObjective, 773
  - PoissonGaussianNoise2DObjective, 762
- ModelDataT
  - mappel, 28
  - mappel::Gauss1DMAP, 134
  - mappel::Gauss1DMLE, 162
  - mappel::Gauss1DsMAP, 217
  - mappel::Gauss1DsMLE, 245
  - mappel::Gauss2DMAP, 301
  - mappel::Gauss2DMLE, 331
  - mappel::Gauss2DsMAP, 391
  - mappel::Gauss2DsMLE, 423
  - mappel::Gauss2DsxyMAP, 486
  - mappel::PoissonNoise1DObjective, 766
  - mappel::PoissonNoise2DObjective, 773
  - PoissonGaussianNoise2DObjective, 762
- ModelImprovement
  - mappel::estimator, 50
- modified\_cholesky
  - mappel, 40
- mtx
  - mappel::estimator::CGaussHeuristicEstimator, 103
  - mappel::estimator::CGaussMLE, 116
  - mappel::estimator::HeuristicEstimator, 542
  - mappel::estimator::IterativeMaximizer, 575
  - mappel::estimator::NewtonDiagonalMaximizer, 718
  - mappel::estimator::NewtonMaximizer, 739
  - mappel::estimator::QuasiNewtonMaximizer, 803
  - mappel::estimator::SimulatedAnnealingMaximizer, 816
  - mappel::estimator::ThreadedEstimator, 852
  - mappel::estimator::TrustRegionMaximizer, 873
- nBacktracks
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 599
- nIterations
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 599
- name
  - mappel::Gauss1DMAP, 154
  - mappel::Gauss1DMLE, 182
  - mappel::Gauss1DsMAP, 237
  - mappel::Gauss1DsMLE, 265
  - mappel::Gauss2DMAP, 322
  - mappel::Gauss2DMLE, 352
  - mappel::Gauss2DsMAP, 414
  - mappel::Gauss2DsMLE, 446
  - mappel::Gauss2DsxyMAP, 504
  - mappel::estimator::CGaussHeuristicEstimator, 102
  - mappel::estimator::CGaussMLE, 114
  - mappel::estimator::Estimator, 126
  - mappel::estimator::HeuristicEstimator, 540
  - mappel::estimator::IterativeMaximizer, 571
  - mappel::estimator::NewtonDiagonalMaximizer, 714
  - mappel::estimator::NewtonMaximizer, 735
  - mappel::estimator::QuasiNewtonMaximizer, 799
  - mappel::estimator::SimulatedAnnealingMaximizer, 814
  - mappel::estimator::ThreadedEstimator, 850
  - mappel::estimator::TrustRegionMaximizer, 869
- Nburnin
  - mappel::mcmc::MCMCData, 678
  - mappel::mcmc::MCMCDataStack, 680
- Ndata
  - mappel::estimator::ProfileBoundsDataStack, 782
  - mappel::mcmc::MCMCDataStack, 681
- negative\_definite\_hessian
  - mappel::methods::objective, 76, 77
- negative\_definite\_hessian\_stack
  - mappel::methods::objective::openmp, 83

- NewtonDiagonalMaximizer
  - mappel::estimator::NewtonDiagonalMaximizer, [702](#)
- NewtonMaximizer
  - mappel::estimator::NewtonMaximizer, [723](#)
- norm\_sq
  - mappel, [40](#)
- normal\_prior\_grad
  - mappel, [40](#)
- normal\_prior\_grad2
  - mappel, [40](#)
- normal\_quantile\_onesided
  - mappel, [40](#)
- normal\_quantile\_twosided
  - mappel, [41](#)
- NotImplementedError
  - mappel::NotImplementedError, [742](#)
- Nparams\_est
  - mappel::estimator::ProfileBoundsData, [778](#)
  - mappel::estimator::ProfileBoundsDataStack, [782](#)
- Nsample
  - mappel::mcmc::MCMCData, [678](#)
  - mappel::mcmc::MCMCDataStack, [681](#)
  - mappel::mcmc::MCMCDebugData, [683](#)
- num\_dim
  - mappel::Gauss1DMAP, [154](#)
  - mappel::Gauss1DMLE, [182](#)
  - mappel::Gauss1DModel, [208](#)
  - mappel::Gauss1DsMAP, [237](#)
  - mappel::Gauss1DsMLE, [265](#)
  - mappel::Gauss1DsModel, [293](#)
  - mappel::Gauss2DMAP, [322](#)
  - mappel::Gauss2DMLE, [352](#)
  - mappel::Gauss2DModel, [381](#)
  - mappel::Gauss2DsMAP, [414](#)
  - mappel::Gauss2DsMLE, [446](#)
  - mappel::Gauss2DsModel, [477](#)
  - mappel::Gauss2DsxyMAP, [504](#)
  - mappel::Gauss2DsxyModel, [528](#)
  - mappel::ImageFormat1DBase, [548](#)
  - mappel::ImageFormat2DBase, [555](#)
  - mappel::PoissonNoise1DObjective, [769](#)
  - mappel::PoissonNoise2DObjective, [776](#)
- num\_estimations
  - mappel::estimator::CGaussHeuristicEstimator, [104](#)
  - mappel::estimator::CGaussMLE, [116](#)
  - mappel::estimator::Estimator, [128](#)
  - mappel::estimator::HeuristicEstimator, [542](#)
  - mappel::estimator::IterativeMaximizer, [575](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [718](#)
  - mappel::estimator::NewtonMaximizer, [739](#)
  - mappel::estimator::QuasiNewtonMaximizer, [803](#)
  - mappel::estimator::SimulatedAnnealingMaximizer, [816](#)
  - mappel::estimator::ThreadedEstimator, [852](#)
  - mappel::estimator::TrustRegionMaximizer, [874](#)
- num\_fixed\_parameters
  - mappel::estimator::IterativeMaximizer::Maximizer↵Data, [595](#)
- num\_hyperparams
  - mappel::Gauss1DMAP, [154](#)
  - mappel::Gauss1DMLE, [182](#)
  - mappel::Gauss1DModel, [209](#)
  - mappel::Gauss1DsMAP, [237](#)
  - mappel::Gauss1DsMLE, [265](#)
  - mappel::Gauss1DsModel, [293](#)
  - mappel::Gauss2DMAP, [322](#)
  - mappel::Gauss2DMLE, [352](#)
  - mappel::Gauss2DModel, [381](#)
  - mappel::Gauss2DsMAP, [414](#)
  - mappel::Gauss2DsMLE, [446](#)
  - mappel::Gauss2DsModel, [477](#)
  - mappel::Gauss2DsxyMAP, [505](#)
  - mappel::Gauss2DsxyModel, [528](#)
  - mappel::MAPEstimator, [591](#)
  - mappel::MCMCAdaptor1Ds, [636](#)
  - mappel::MCMCAdaptor1D, [618](#)
  - mappel::MCMCAdaptor2Ds, [673](#)
  - mappel::MCMCAdaptor2D, [654](#)
  - mappel::MLEstimator, [697](#)
  - mappel::PointEmitterModel, [759](#)
- num\_oversample
  - mappel::mcmc, [55](#)
- num\_params
  - mappel::Gauss1DMAP, [154](#)
  - mappel::Gauss1DMLE, [182](#)
  - mappel::Gauss1DModel, [209](#)
  - mappel::Gauss1DsMAP, [237](#)
  - mappel::Gauss1DsMLE, [266](#)
  - mappel::Gauss1DsModel, [293](#)
  - mappel::Gauss2DMAP, [322](#)
  - mappel::Gauss2DMLE, [352](#)
  - mappel::Gauss2DModel, [381](#)
  - mappel::Gauss2DsMAP, [414](#)
  - mappel::Gauss2DsMLE, [446](#)
  - mappel::Gauss2DsModel, [477](#)
  - mappel::Gauss2DsxyMAP, [505](#)
  - mappel::Gauss2DsxyModel, [528](#)
  - mappel::MAPEstimator, [591](#)
  - mappel::MCMCAdaptor1Ds, [636](#)
  - mappel::MCMCAdaptor1D, [618](#)
  - mappel::MCMCAdaptor2Ds, [673](#)
  - mappel::MCMCAdaptor2D, [654](#)
  - mappel::MLEstimator, [697](#)
  - mappel::PointEmitterModel, [760](#)
  - mappel::estimator::IterativeMaximizer::Maximizer↵Data, [600](#)
- num\_phases
  - mappel::Gauss1DMAP, [154](#)

- mappel::Gauss1DMLE, 182
- mappel::Gauss1DModel, 209
- mappel::Gauss1DsMAP, 238
- mappel::Gauss1DsMLE, 266
- mappel::Gauss1DsModel, 293
- mappel::Gauss2DMAP, 323
- mappel::Gauss2DMLE, 353
- mappel::Gauss2DModel, 382
- mappel::Gauss2DsMAP, 414
- mappel::Gauss2DsMLE, 446
- mappel::Gauss2DsModel, 478
- mappel::Gauss2DsxyMAP, 505
- mappel::MCMCAdaptor1Ds, 636
- mappel::MCMCAdaptor1D, 618
- mappel::MCMCAdaptor2Ds, 673
- mappel::MCMCAdaptor2D, 654
- mappel::MCMCAdaptorBase, 676
- num\_threads
  - mappel::estimator::CGaussHeuristicEstimator, 104
  - mappel::estimator::CGaussMLE, 116
  - mappel::estimator::HeuristicEstimator, 542
  - mappel::estimator::IterativeMaximizer, 575
  - mappel::estimator::NewtonDiagonalMaximizer, 719
  - mappel::estimator::NewtonMaximizer, 740
  - mappel::estimator::QuasiNewtonMaximizer, 804
  - mappel::estimator::SimulatedAnnealingMaximizer, 817
  - mappel::estimator::ThreadedEstimator, 852
  - mappel::estimator::TrustRegionMaximizer, 874
- numerical.cpp, 921
- numerical.h, 922
- NumericalError
  - mappel::NumericalError, 742
- OMPEXceptionCatcher
  - omp\_exception\_catcher, 90
  - omp\_exception\_catcher::impl\_::OMPEXceptionCatcher, 743
- OMPEXceptionCatcher.h, 923
- observed\_information
  - mappel::methods, 64
- omp\_exception\_catcher, 90
  - Abort, 91
  - Continue, 91
  - DoNotTry, 91
  - OMPEXceptionCatcher, 90
  - RethrowFirst, 91
  - Strategy, 91
- omp\_exception\_catcher::impl\_, 91
- omp\_exception\_catcher::impl\_::OMPEXceptionCatcher
  - OMPEXceptionCatcher, 743
  - rethrow, 744
  - run, 744
  - setGlobalDefaultStrategy, 744
- omp\_exception\_catcher::impl\_::OMPEXceptionCatcher<\_dummy >, 743
- openmp\_methods.h, 925
- operator<<
  - mappel, 41
  - mappel::Gauss1DModel::Stencil, 839
  - mappel::Gauss1DsModel::Stencil, 825
  - mappel::Gauss2DModel::Stencil, 829
  - mappel::Gauss2DsModel::Stencil, 834
  - mappel::Gauss2DsxyModel::Stencil, 820
  - mappel::estimator, 50
  - mappel::estimator::Estimator, 127
- operator=
  - mappel::Gauss1DMAP, 144, 145
  - mappel::Gauss1DMLE, 172, 173
  - mappel::Gauss1DModel, 199, 200
  - mappel::Gauss1DsMAP, 227
  - mappel::Gauss1DsMLE, 255, 256
  - mappel::Gauss1DsModel, 283, 284
  - mappel::Gauss2DMAP, 313
  - mappel::Gauss2DMLE, 343
  - mappel::Gauss2DModel, 372
  - mappel::Gauss2DsMAP, 403
  - mappel::Gauss2DsMLE, 435, 436
  - mappel::Gauss2DsModel, 467
  - mappel::ImageFormat2DBase, 554
  - mappel::MAPEstimator, 586
  - mappel::MCMCAdaptor1Ds, 628, 629
  - mappel::MCMCAdaptor1D, 611
  - mappel::MCMCAdaptor2Ds, 665
  - mappel::MCMCAdaptor2D, 646, 647
  - mappel::MLEstimator, 692
  - mappel::PointEmitterModel, 755
  - mappel::PoissonNoise1DObjective, 768
  - mappel::PoissonNoise2DObjective, 775
- ParallelRngGeneratorT
  - mappel, 28
- ParallelRngManagerT
  - mappel, 28
- ParamVecT
  - mappel, 28
  - mappel::Gauss1DMAP, 134
  - mappel::Gauss1DMLE, 162
  - mappel::Gauss1DModel, 190
  - mappel::Gauss1DsMAP, 217
  - mappel::Gauss1DsMLE, 245
  - mappel::Gauss1DsModel, 273
  - mappel::Gauss2DMAP, 302
  - mappel::Gauss2DMLE, 332
  - mappel::Gauss2DModel, 361
  - mappel::Gauss2DsMAP, 391
  - mappel::Gauss2DsMLE, 423
  - mappel::Gauss2DsModel, 455

- mappel::Gauss2DsxyMAP, 486
- mappel::Gauss2DsxyModel, 512
- mappel::MAPEstimator, 580
- mappel::MCMCAdaptor1Ds, 622
- mappel::MCMCAdaptor1D, 604
- mappel::MCMCAdaptor2Ds, 658
- mappel::MCMCAdaptor2D, 640
- mappel::MLEstimator, 686
- mappel::PointEmitterModel, 749
- ParameterValueError
  - mappel::ParameterValueError, 745
- ParamT
  - mappel, 28
  - mappel::Gauss1DMAP, 134
  - mappel::Gauss1DMLE, 162
  - mappel::Gauss1DModel, 190
  - mappel::Gauss1DModel::Stencil, 837
  - mappel::Gauss1DsMAP, 217
  - mappel::Gauss1DsMLE, 245
  - mappel::Gauss1DsModel, 273
  - mappel::Gauss1DsModel::Stencil, 823
  - mappel::Gauss2DMAP, 301
  - mappel::Gauss2DMLE, 331
  - mappel::Gauss2DModel, 361
  - mappel::Gauss2DModel::Stencil, 827
  - mappel::Gauss2DsMAP, 391
  - mappel::Gauss2DsMLE, 423
  - mappel::Gauss2DsModel, 455
  - mappel::Gauss2DsModel::Stencil, 832
  - mappel::Gauss2DsxyMAP, 486
  - mappel::Gauss2DsxyModel, 512
  - mappel::Gauss2DsxyModel::Stencil, 818
  - mappel::MAPEstimator, 580
  - mappel::MCMCAdaptor1Ds, 622
  - mappel::MCMCAdaptor1D, 604
  - mappel::MCMCAdaptor2Ds, 658
  - mappel::MCMCAdaptor2D, 640
  - mappel::MLEstimator, 686
  - mappel::PointEmitterModel, 749
- pareto\_prior\_grad
  - mappel, 42
- pareto\_prior\_grad2
  - mappel, 42
- pixel\_grad
  - mappel::Gauss1DMAP, 145
  - mappel::Gauss1DMLE, 173
  - mappel::Gauss1DModel, 200
  - mappel::Gauss1DsMAP, 228
  - mappel::Gauss1DsMLE, 256
  - mappel::Gauss1DsModel, 284
  - mappel::Gauss2DMAP, 313
  - mappel::Gauss2DMLE, 343
  - mappel::Gauss2DModel, 372
  - mappel::Gauss2DsMAP, 404
  - mappel::Gauss2DsMLE, 436
  - mappel::Gauss2DsModel, 467
  - mappel::Gauss2DsxyMAP, 496
  - mappel::Gauss2DsxyModel, 522
- pixel\_grad2
  - mappel::Gauss1DMAP, 145
  - mappel::Gauss1DMLE, 173
  - mappel::Gauss1DModel, 200
  - mappel::Gauss1DsMAP, 228
  - mappel::Gauss1DsMLE, 256
  - mappel::Gauss1DsModel, 284
  - mappel::Gauss2DMAP, 313
  - mappel::Gauss2DMLE, 343
  - mappel::Gauss2DModel, 372
  - mappel::Gauss2DsMAP, 404
  - mappel::Gauss2DsMLE, 436
  - mappel::Gauss2DsModel, 467
  - mappel::Gauss2DsxyMAP, 496
  - mappel::Gauss2DsxyModel, 522
- pixel\_hess
  - mappel::Gauss1DMAP, 145
  - mappel::Gauss1DMLE, 173
  - mappel::Gauss1DModel, 200
  - mappel::Gauss1DsMAP, 228
  - mappel::Gauss1DsMLE, 256
  - mappel::Gauss1DsModel, 284
  - mappel::Gauss2DMAP, 313
  - mappel::Gauss2DMLE, 343
  - mappel::Gauss2DModel, 373
  - mappel::Gauss2DsMAP, 404
  - mappel::Gauss2DsMLE, 436
  - mappel::Gauss2DsModel, 468
  - mappel::Gauss2DsxyMAP, 496
  - mappel::Gauss2DsxyModel, 522
- pixel\_hess\_update
  - mappel::Gauss1DMAP, 145
  - mappel::Gauss1DMLE, 173
  - mappel::Gauss1DModel, 200
  - mappel::Gauss1DsMAP, 228
  - mappel::Gauss1DsMLE, 256
  - mappel::Gauss1DsModel, 284
  - mappel::Gauss2DMAP, 314
  - mappel::Gauss2DMLE, 344
  - mappel::Gauss2DModel, 373
  - mappel::Gauss2DsMAP, 404
  - mappel::Gauss2DsMLE, 436
  - mappel::Gauss2DsModel, 468
  - mappel::Gauss2DsxyMAP, 496
  - mappel::Gauss2DsxyModel, 522
- pixel\_model\_value
  - mappel::Gauss1DMAP, 145
  - mappel::Gauss1DMLE, 173
  - mappel::Gauss1DModel, 200
  - mappel::Gauss1DsMAP, 228

- mappel::Gauss1DsMLE, [256](#)
- mappel::Gauss1DsModel, [284](#)
- mappel::Gauss2DMap, [314](#)
- mappel::Gauss2DMLE, [344](#)
- mappel::Gauss2DModel, [373](#)
- mappel::Gauss2DsMap, [404](#)
- mappel::Gauss2DsMLE, [437](#)
- mappel::Gauss2DsModel, [468](#)
- mappel::Gauss2DsxyMap, [496](#)
- mappel::Gauss2DsxyModel, [522](#)
- PointEmitterModel
  - mappel::PointEmitterModel, [749](#)
- PointEmitterModel.cpp, [927](#)
- PointEmitterModel.h, [928](#)
- poisson\_log\_likelihood
  - mappel, [42](#)
- PoissonGaussianNoise2DObjective
  - CoordIdxT, [761](#)
  - CoordStackT, [761](#)
  - CoordT, [761](#)
  - estimator\_names, [762](#)
  - ModelDataStackT, [762](#)
  - ModelDataT, [762](#)
  - PoissonGaussianNoise2DObjective, [762](#)
  - sensor\_bg\_map, [762](#)
  - sensor\_gain\_map, [762](#)
- PoissonGaussianNoise2DObjective< ModelBase >, [760](#)
- PoissonGaussianNoise2DObjective.cpp, [929](#)
- PoissonGaussianNoise2DObjective.h, [929](#)
  - fisher\_information, [931](#)
  - log\_likelihood, [931](#)
  - make\_estimator, [931](#)
  - model\_grad, [931](#)
  - model\_grad2, [931](#)
  - model\_hessian, [931](#)
  - model\_image, [931](#)
  - relative\_log\_likelihood, [931](#)
  - simulate\_image, [932](#)
- PoissonNoise1DObjective
  - mappel::PoissonNoise1DObjective, [766](#)
- PoissonNoise1DObjective.cpp, [932](#)
- PoissonNoise1DObjective.h, [933](#)
- PoissonNoise2DObjective
  - mappel::PoissonNoise2DObjective, [773](#)
- PoissonNoise2DObjective.cpp, [935](#)
- PoissonNoise2DObjective.h, [935](#)
- print\_centered\_title
  - mappel, [42](#)
- print\_image
  - mappel, [42](#)
- print\_labeled\_image
  - mappel, [43](#)
- print\_text\_image
  - mappel, [43](#)
- print\_vec\_row
  - mappel, [43](#)
- prior
  - mappel::Gauss1DMap, [154](#)
  - mappel::Gauss1DMLE, [182](#)
  - mappel::Gauss1DModel, [209](#)
  - mappel::Gauss1DsMap, [238](#)
  - mappel::Gauss1DsMLE, [266](#)
  - mappel::Gauss1DsModel, [294](#)
  - mappel::Gauss2DMap, [323](#)
  - mappel::Gauss2DMLE, [353](#)
  - mappel::Gauss2DModel, [382](#)
  - mappel::Gauss2DsMap, [415](#)
  - mappel::Gauss2DsMLE, [447](#)
  - mappel::Gauss2DsModel, [478](#)
  - mappel::Gauss2DsxyMap, [505](#)
  - mappel::Gauss2DsxyModel, [529](#)
  - mappel::MAPEstimator, [591](#)
  - mappel::MCMCAdaptor1Ds, [636](#)
  - mappel::MCMCAdaptor1D, [618](#)
  - mappel::MCMCAdaptor2Ds, [673](#)
  - mappel::MCMCAdaptor2D, [654](#)
  - mappel::MLEstimator, [697](#)
  - mappel::PointEmitterModel, [760](#)
- prior\_objective
  - mappel::methods, [64](#)
- prior\_types
  - mappel::Gauss1DMap, [155](#)
  - mappel::Gauss1DMLE, [183](#)
  - mappel::Gauss1DModel, [209](#)
  - mappel::Gauss1DsMap, [238](#)
  - mappel::Gauss1DsMLE, [266](#)
  - mappel::Gauss1DsModel, [294](#)
  - mappel::Gauss2DMap, [323](#)
  - mappel::Gauss2DMLE, [353](#)
  - mappel::Gauss2DModel, [382](#)
  - mappel::Gauss2DsMap, [415](#)
  - mappel::Gauss2DsMLE, [447](#)
  - mappel::Gauss2DsModel, [478](#)
  - mappel::Gauss2DsxyMap, [505](#)
- profile\_bound\_backtrack
  - mappel::estimator::IterativeMaximizer, [571](#)
  - mappel::estimator::NewtonDiagonalMaximizer, [714](#)
  - mappel::estimator::NewtonMaximizer, [735](#)
  - mappel::estimator::QuasiNewtonMaximizer, [799](#)
  - mappel::estimator::TrustRegionMaximizer, [869](#)
- profile\_lb
  - mappel::estimator::ProfileBoundsData, [779](#)
  - mappel::estimator::ProfileBoundsDataStack, [782](#)
- profile\_points\_lb
  - mappel::estimator::ProfileBoundsData, [779](#)
  - mappel::estimator::ProfileBoundsDataStack, [782](#)
- profile\_points\_lb\_rllh
  - mappel::estimator::ProfileBoundsData, [779](#)



- mappel::estimator::ProfileBoundsDataStack, 783
- profile\_points\_ub
  - mappel::estimator::ProfileBoundsData, 779
  - mappel::estimator::ProfileBoundsDataStack, 783
- profile\_points\_ub\_rllh
  - mappel::estimator::ProfileBoundsData, 779
  - mappel::estimator::ProfileBoundsDataStack, 783
- profile\_ub
  - mappel::estimator::ProfileBoundsData, 780
  - mappel::estimator::ProfileBoundsDataStack, 783
- psf\_sigma
  - mappel::Gauss1DMAP, 155
  - mappel::Gauss1DMLE, 183
  - mappel::Gauss1DModel, 210
  - mappel::Gauss2DMAP, 323
  - mappel::Gauss2DMLE, 353
  - mappel::Gauss2DModel, 382
  - mappel::Gauss2DsxyMAP, 506
- QuasiNewtonMaximizer
  - mappel::estimator::QuasiNewtonMaximizer, 787
- README.md, 937
- record\_backtrack
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 595
- record\_exit\_code
  - mappel::estimator::CGaussHeuristicEstimator, 102
  - mappel::estimator::CGaussMLE, 114
  - mappel::estimator::Estimator, 126
  - mappel::estimator::HeuristicEstimator, 540
  - mappel::estimator::IterativeMaximizer, 571
  - mappel::estimator::NewtonDiagonalMaximizer, 714
  - mappel::estimator::NewtonMaximizer, 735
  - mappel::estimator::QuasiNewtonMaximizer, 799
  - mappel::estimator::SimulatedAnnealingMaximizer, 814
  - mappel::estimator::ThreadedEstimator, 851
  - mappel::estimator::TrustRegionMaximizer, 869
- record\_iteration
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 596
- record\_run\_statistics
  - mappel::estimator::IterativeMaximizer, 571
  - mappel::estimator::NewtonDiagonalMaximizer, 715
  - mappel::estimator::NewtonMaximizer, 736
  - mappel::estimator::QuasiNewtonMaximizer, 800
  - mappel::estimator::TrustRegionMaximizer, 870
- record\_walltime
  - mappel::estimator::CGaussHeuristicEstimator, 102
  - mappel::estimator::CGaussMLE, 115
  - mappel::estimator::Estimator, 127
  - mappel::estimator::HeuristicEstimator, 541
  - mappel::estimator::IterativeMaximizer, 572
  - mappel::estimator::NewtonDiagonalMaximizer, 715
  - mappel::estimator::NewtonMaximizer, 736
  - mappel::estimator::QuasiNewtonMaximizer, 800
  - mappel::estimator::SimulatedAnnealingMaximizer, 815
  - mappel::estimator::ThreadedEstimator, 851
  - mappel::estimator::TrustRegionMaximizer, 870
- refine\_gaussian\_2Dmax
  - mappel, 43
- refine\_gaussian\_3Dmax
  - mappel, 43
- reflected\_theta
  - mappel::Gauss1DMAP, 146
  - mappel::Gauss1DMLE, 174
  - mappel::Gauss1DModel, 200
  - mappel::Gauss1DsMAP, 228
  - mappel::Gauss1DsMLE, 257
  - mappel::Gauss1DsModel, 285
  - mappel::Gauss2DMAP, 314
  - mappel::Gauss2DMLE, 344
  - mappel::Gauss2DModel, 373
  - mappel::Gauss2DsMAP, 405
  - mappel::Gauss2DsMLE, 437
  - mappel::Gauss2DsModel, 468
  - mappel::Gauss2DsxyMAP, 496
  - mappel::Gauss2DsxyModel, 522
  - mappel::MAPEstimator, 586
  - mappel::MCMCAdaptor1Ds, 629
  - mappel::MCMCAdaptor1D, 611
  - mappel::MCMCAdaptor2Ds, 665
  - mappel::MCMCAdaptor2D, 647
  - mappel::MLEstimator, 692
  - mappel::PointEmitterModel, 755
- reflected\_theta\_stack
  - mappel::Gauss1DMAP, 146
  - mappel::Gauss1DMLE, 174
  - mappel::Gauss1DModel, 201
  - mappel::Gauss1DsMAP, 229
  - mappel::Gauss1DsMLE, 257
  - mappel::Gauss1DsModel, 285
  - mappel::Gauss2DMAP, 314
  - mappel::Gauss2DMLE, 344
  - mappel::Gauss2DModel, 373
  - mappel::Gauss2DsMAP, 405
  - mappel::Gauss2DsMLE, 437
  - mappel::Gauss2DsModel, 468
  - mappel::Gauss2DsxyMAP, 497
  - mappel::Gauss2DsxyModel, 522
  - mappel::MAPEstimator, 586
  - mappel::MCMCAdaptor1Ds, 629
  - mappel::MCMCAdaptor1D, 611
  - mappel::MCMCAdaptor2Ds, 666
  - mappel::MCMCAdaptor2D, 647
  - mappel::MLEstimator, 692
  - mappel::PointEmitterModel, 755

- relative\_log\_likelihood
  - PoissonGaussianNoise2DObjective.h, [931](#)
- relative\_poisson\_log\_likelihood
  - mappel, [44](#)
- rename\_hyperparam
  - mappel::Gauss1DModel, [146](#)
  - mappel::Gauss1DModel, [174](#)
  - mappel::Gauss1DModel, [201](#)
  - mappel::Gauss1DsMAP, [229](#)
  - mappel::Gauss1DsMLE, [257](#)
  - mappel::Gauss1DsModel, [285](#)
  - mappel::Gauss2DModel, [314](#)
  - mappel::Gauss2DModel, [344](#)
  - mappel::Gauss2DModel, [373](#)
  - mappel::Gauss2DsMAP, [405](#)
  - mappel::Gauss2DsMLE, [437](#)
  - mappel::Gauss2DsModel, [469](#)
  - mappel::Gauss2DsxyMAP, [497](#)
  - mappel::Gauss2DsxyModel, [523](#)
  - mappel::MAPEstimator, [586](#)
  - mappel::MCMCAdaptor1Ds, [629](#)
  - mappel::MCMCAdaptor1D, [611](#)
  - mappel::MCMCAdaptor2Ds, [666](#)
  - mappel::MCMCAdaptor2D, [647](#)
  - mappel::MLEstimator, [692](#)
  - mappel::PointEmitterModel, [755](#)
- restore\_stencil
  - mappel::estimator::IterativeMaximizer::Maximizer↔  
Data, [596](#)
- restrict\_value\_range
  - mappel, [44](#)
- rethrow
  - omp\_exception\_catcher::impl\_::OMPEXception↔  
Catcher, [744](#)
- RethrowFirst
  - omp\_exception\_catcher, [91](#)
- ReturnIfSubclassT
  - mappel, [28](#)
- rho\_cauchy\_min
  - mappel::estimator::TrustRegionMaximizer, [874](#)
- rho\_obj\_min
  - mappel::estimator::TrustRegionMaximizer, [874](#)
- rho\_obj\_opt
  - mappel::estimator::TrustRegionMaximizer, [874](#)
- rllh
  - mappel::estimator::IterativeMaximizer::Maximizer↔  
Data, [600](#)
  - mappel::methods::likelihood, [70](#)
  - mappel::methods::objective, [77](#)
- rllh\_beta2\_prior
  - mappel, [44](#)
- rllh\_beta\_prior
  - mappel, [44](#)
- rllh\_components
  - mappel::methods::likelihood::debug, [72](#)
  - mappel::methods::objective::debug, [80](#)
- rllh\_gamma\_prior
  - mappel, [44](#)
- rllh\_normal\_prior
  - mappel, [44](#)
- rllh\_pareto\_prior
  - mappel, [44](#)
- rllh\_stack
  - mappel::methods::objective::openmp, [83](#), [84](#)
- rng.cpp, [937](#)
- rng.h, [938](#)
- rng\_manager
  - mappel, [45](#)
- RngSeedT
  - mappel, [28](#)
- run
  - omp\_exception\_catcher::impl\_::OMPEXception↔  
Catcher, [744](#)
- s0
  - mappel::estimator::IterativeMaximizer::Maximizer↔  
Data, [600](#)
- s1
  - mappel::estimator::IterativeMaximizer::Maximizer↔  
Data, [600](#)
- sample
  - mappel::mcmc::MCMCData, [678](#)
  - mappel::mcmc::MCMCDataStack, [681](#)
  - mappel::mcmc::MCMCDebugData, [683](#)
- sample\_cov
  - mappel::mcmc::MCMCData, [679](#)
  - mappel::mcmc::MCMCDataStack, [681](#)
- sample\_mcmc\_candidate
  - mappel::Gauss1DModel, [146](#)
  - mappel::Gauss1DModel, [174](#)
  - mappel::Gauss1DModel, [201](#)
  - mappel::Gauss1DsMAP, [229](#)
  - mappel::Gauss1DsMLE, [257](#)
  - mappel::Gauss1DsModel, [285](#)
  - mappel::Gauss2DModel, [314](#), [315](#)
  - mappel::Gauss2DModel, [344](#), [345](#)
  - mappel::Gauss2DModel, [374](#)
  - mappel::Gauss2DsMAP, [405](#)
  - mappel::Gauss2DsMLE, [437](#), [438](#)
  - mappel::Gauss2DsModel, [469](#)
  - mappel::Gauss2DsxyMAP, [497](#)
  - mappel::Gauss2DsxyModel, [523](#)
  - mappel::MCMCAdaptor1Ds, [629](#)
  - mappel::MCMCAdaptor1D, [611](#), [612](#)
  - mappel::MCMCAdaptor2Ds, [666](#)
  - mappel::MCMCAdaptor2D, [647](#)
- sample\_mean
  - mappel::mcmc::MCMCData, [679](#)



- mappel::mcmc::MCMCDataStack, 681
- sample\_posterior
  - mappel::mcmc, 55
- sample\_posterior\_debug
  - mappel::mcmc, 55
- sample\_prior
  - mappel::Gauss1DMap, 146, 147
  - mappel::Gauss1DMLE, 174, 175
  - mappel::Gauss1DModel, 201
  - mappel::Gauss1DsMAP, 229
  - mappel::Gauss1DsMLE, 257, 258
  - mappel::Gauss1DsModel, 285, 286
  - mappel::Gauss2DMap, 315
  - mappel::Gauss2DMLE, 345
  - mappel::Gauss2DModel, 374
  - mappel::Gauss2DsMAP, 405, 406
  - mappel::Gauss2DsMLE, 438
  - mappel::Gauss2DsModel, 469
  - mappel::Gauss2DsxyMAP, 497
  - mappel::Gauss2DsxyModel, 523
  - mappel::MAPEstimator, 586, 587
  - mappel::MCMCAdaptor1Ds, 630
  - mappel::MCMCAdaptor1D, 612
  - mappel::MCMCAdaptor2Ds, 666
  - mappel::MCMCAdaptor2D, 648
  - mappel::MLEstimator, 692, 693
  - mappel::PointEmitterModel, 755, 756
- sample\_prior\_stack
  - mappel::methods::openmp, 89
- sample\_rllh
  - mappel::mcmc::MCMCData, 679
  - mappel::mcmc::MCMCDataStack, 681
  - mappel::mcmc::MCMCDebugData, 683
- save\_stencil
  - mappel::estimator::IterativeMaximizer::Maximizer↵  
Data, 596
- saved\_stencil
  - mappel::estimator::IterativeMaximizer::Maximizer↵  
Data, 597
- saved\_theta
  - mappel::estimator::IterativeMaximizer::Maximizer↵  
Data, 597
- sensor\_bg\_map
  - PoissonGaussianNoise2DObjective, 762
- sensor\_gain\_map
  - PoissonGaussianNoise2DObjective, 762
- seq\_len
  - mappel::estimator::IterativeMaximizer::Maximizer↵  
Data, 600
- seq\_rllh
  - mappel::estimator::IterativeMaximizer::Maximizer↵  
Data, 600
- set\_background\_mcmc\_sampling
  - mappel::Gauss1DMap, 147
  - mappel::Gauss1DMLE, 175
  - mappel::Gauss1DModel, 202
  - mappel::Gauss1DsMAP, 230
  - mappel::Gauss1DsMLE, 258
  - mappel::Gauss1DsModel, 286
  - mappel::Gauss2DMap, 315
  - mappel::Gauss2DMLE, 345
  - mappel::Gauss2DModel, 374
  - mappel::Gauss2DsMAP, 406
  - mappel::Gauss2DsMLE, 438
  - mappel::Gauss2DsModel, 470
- set\_bounds
  - mappel::Gauss1DMap, 147
  - mappel::Gauss1DMLE, 175
  - mappel::Gauss1DModel, 202
  - mappel::Gauss1DsMAP, 230
  - mappel::Gauss1DsMLE, 258
  - mappel::Gauss1DsModel, 286
  - mappel::Gauss2DMap, 315
  - mappel::Gauss2DMLE, 345
  - mappel::Gauss2DModel, 374
  - mappel::Gauss2DsMAP, 406
  - mappel::Gauss2DsMLE, 438
  - mappel::Gauss2DsModel, 470
  - mappel::Gauss2DsxyMAP, 498
  - mappel::Gauss2DsxyModel, 523
  - mappel::MAPEstimator, 587
  - mappel::MCMCAdaptor1Ds, 630
  - mappel::MCMCAdaptor1D, 612
  - mappel::MCMCAdaptor2Ds, 667
  - mappel::MCMCAdaptor2D, 648
  - mappel::MLEstimator, 693
  - mappel::PointEmitterModel, 756
- set\_fixed\_parameters
  - mappel::estimator::IterativeMaximizer::Maximizer↵  
Data, 597
- set\_hyperparam\_names
  - mappel::Gauss1DMap, 147
  - mappel::Gauss1DMLE, 175
  - mappel::Gauss1DModel, 202
  - mappel::Gauss1DsMAP, 230
  - mappel::Gauss1DsMLE, 258
  - mappel::Gauss1DsModel, 286
  - mappel::Gauss2DMap, 315
  - mappel::Gauss2DMLE, 345
  - mappel::Gauss2DModel, 375
  - mappel::Gauss2DsMAP, 406
  - mappel::Gauss2DsMLE, 438
  - mappel::Gauss2DsModel, 470

- mappel::Gauss2DsxyMAP, [498](#)
- mappel::Gauss2DsxyModel, [523](#)
- mappel::MAPEstimator, [587](#)
- mappel::MCMCAdaptor1Ds, [630](#)
- mappel::MCMCAdaptor1D, [612](#)
- mappel::MCMCAdaptor2Ds, [667](#)
- mappel::MCMCAdaptor2D, [648](#)
- mappel::MLEstimator, [693](#)
- mappel::PointEmitterModel, [756](#)
- set\_hyperparam\_value
  - mappel::Gauss1DMAP, [147](#)
  - mappel::Gauss1DMLE, [175](#)
  - mappel::Gauss1DModel, [202](#)
  - mappel::Gauss1DsMAP, [230](#)
  - mappel::Gauss1DsMLE, [258](#)
  - mappel::Gauss1DsModel, [286](#)
  - mappel::Gauss2DMAP, [316](#)
  - mappel::Gauss2DMLE, [346](#)
  - mappel::Gauss2DModel, [375](#)
  - mappel::Gauss2DsMAP, [406](#)
  - mappel::Gauss2DsMLE, [439](#)
  - mappel::Gauss2DsModel, [470](#)
  - mappel::Gauss2DsxyMAP, [498](#)
  - mappel::Gauss2DsxyModel, [523](#)
  - mappel::MAPEstimator, [587](#)
  - mappel::MCMCAdaptor1Ds, [630](#)
  - mappel::MCMCAdaptor1D, [613](#)
  - mappel::MCMCAdaptor2Ds, [667](#)
  - mappel::MCMCAdaptor2D, [648](#)
  - mappel::MLEstimator, [693](#)
  - mappel::PointEmitterModel, [756](#)
- set\_hyperparams
  - mappel::Gauss1DMAP, [147](#)
  - mappel::Gauss1DMLE, [175](#)
  - mappel::Gauss1DModel, [202](#)
  - mappel::Gauss1DsMAP, [230](#)
  - mappel::Gauss1DsMLE, [258](#)
  - mappel::Gauss1DsModel, [286](#)
  - mappel::Gauss2DMAP, [316](#)
  - mappel::Gauss2DMLE, [346](#)
  - mappel::Gauss2DModel, [375](#)
  - mappel::Gauss2DsMAP, [406](#)
  - mappel::Gauss2DsMLE, [439](#)
  - mappel::Gauss2DsModel, [470](#)
  - mappel::Gauss2DsxyMAP, [498](#)
  - mappel::Gauss2DsxyModel, [524](#)
  - mappel::MAPEstimator, [587](#)
  - mappel::MCMCAdaptor1Ds, [631](#)
  - mappel::MCMCAdaptor1D, [613](#)
  - mappel::MCMCAdaptor2Ds, [667](#)
  - mappel::MCMCAdaptor2D, [649](#)
  - mappel::MLEstimator, [693](#)
  - mappel::PointEmitterModel, [756](#)
- set\_image\_in\_stack
  - mappel::Gauss1DMAP, [148](#)
  - mappel::Gauss1DMLE, [176](#)
  - mappel::Gauss1DModel, [202](#)
  - mappel::Gauss1DsMAP, [230](#)
  - mappel::Gauss1DsMLE, [259](#)
  - mappel::Gauss1DsModel, [287](#)
  - mappel::Gauss2DMAP, [316](#)
  - mappel::Gauss2DMLE, [346](#)
  - mappel::Gauss2DModel, [375](#)
  - mappel::Gauss2DsMAP, [407](#)
  - mappel::Gauss2DsMLE, [439](#)
  - mappel::Gauss2DsModel, [470](#)
  - mappel::Gauss2DsxyMAP, [499](#)
  - mappel::Gauss2DsxyModel, [524](#)
  - mappel::MAPEstimator, [587](#)
  - mappel::MCMCAdaptor1Ds, [631](#)
  - mappel::MCMCAdaptor1D, [613](#)
- set\_intensity\_mcmc\_sampling
  - mappel::Gauss1DMAP, [148](#)
  - mappel::Gauss1DMLE, [176](#)
  - mappel::Gauss1DModel, [203](#)
  - mappel::Gauss1DsMAP, [231](#)
  - mappel::Gauss1DsMLE, [259](#)
  - mappel::Gauss1DsModel, [287](#)
  - mappel::Gauss2DMAP, [316](#)
  - mappel::Gauss2DMLE, [346](#)
  - mappel::Gauss2DModel, [375](#)
  - mappel::Gauss2DsMAP, [407](#)
  - mappel::Gauss2DsMLE, [439](#)
  - mappel::Gauss2DsModel, [470](#)
  - mappel::Gauss2DsxyMAP, [499](#)
  - mappel::MCMCAdaptor1Ds, [631](#)
  - mappel::MCMCAdaptor1D, [613](#)
  - mappel::MCMCAdaptor2Ds, [667](#)
  - mappel::MCMCAdaptor2D, [649](#)
- set\_lbound
  - mappel::Gauss1DMAP, [148](#)
  - mappel::Gauss1DMLE, [176](#)
  - mappel::Gauss1DModel, [203](#)
  - mappel::Gauss1DsMAP, [231](#)
  - mappel::Gauss1DsMLE, [259](#)
  - mappel::Gauss1DsModel, [287](#)
  - mappel::Gauss2DMAP, [316](#)
  - mappel::Gauss2DMLE, [346](#)
  - mappel::Gauss2DModel, [375](#)
  - mappel::Gauss2DsMAP, [407](#)
  - mappel::Gauss2DsMLE, [439](#)
  - mappel::Gauss2DsModel, [471](#)
  - mappel::Gauss2DsxyMAP, [499](#)
  - mappel::Gauss2DsxyModel, [524](#)
  - mappel::MAPEstimator, [587](#)
  - mappel::MCMCAdaptor1Ds, [631](#)
  - mappel::MCMCAdaptor1D, [613](#)

- mappel::MCMCAdaptor2Ds, 668
- mappel::MCMCAdaptor2D, 649
- mappel::MLEstimator, 693
- mappel::PointEmitterModel, 756
- set\_max\_sigma
  - mappel::Gauss1DsMAP, 231
  - mappel::Gauss1DsMLE, 259
  - mappel::Gauss1DsModel, 287
  - mappel::Gauss2DsMAP, 407
  - mappel::Gauss2DsMLE, 439
  - mappel::Gauss2DsModel, 471
  - mappel::Gauss2DsxyModel, 524
- set\_max\_sigma\_ratio
  - mappel::Gauss2DsMAP, 407
  - mappel::Gauss2DsMLE, 439
  - mappel::Gauss2DsModel, 471
  - mappel::Gauss2DsxyModel, 524
- set\_mcmc\_num\_phases
  - mappel::Gauss1DMAP, 148
  - mappel::Gauss1DMLE, 176
  - mappel::Gauss1DModel, 203
  - mappel::Gauss1DsMAP, 231
  - mappel::Gauss1DsMLE, 259
  - mappel::Gauss1DsModel, 287
  - mappel::Gauss2DMAP, 316
  - mappel::Gauss2DMLE, 346
  - mappel::Gauss2DModel, 376
  - mappel::Gauss2DsMAP, 407
  - mappel::Gauss2DsMLE, 440
  - mappel::Gauss2DsModel, 471
  - mappel::Gauss2DsxyMAP, 499
  - mappel::MCMCAdaptor1Ds, 631
  - mappel::MCMCAdaptor1D, 613
  - mappel::MCMCAdaptor2Ds, 668
  - mappel::MCMCAdaptor2D, 649
  - mappel::MCMCAdaptorBase, 676
- set\_mcmc\_sigma\_scale
  - mappel::Gauss1DMAP, 148
  - mappel::Gauss1DMLE, 176
  - mappel::Gauss1DModel, 203
  - mappel::Gauss1DsMAP, 231
  - mappel::Gauss1DsMLE, 260
  - mappel::Gauss1DsModel, 288
  - mappel::Gauss2DMAP, 316
  - mappel::Gauss2DMLE, 346
  - mappel::Gauss2DModel, 376
  - mappel::Gauss2DsMAP, 408
  - mappel::Gauss2DsMLE, 440
  - mappel::Gauss2DsModel, 471
  - mappel::Gauss2DsxyMAP, 499
  - mappel::MCMCAdaptor1Ds, 631
  - mappel::MCMCAdaptor1D, 613
  - mappel::MCMCAdaptor2Ds, 668
  - mappel::MCMCAdaptor2D, 649
- mappel::MCMCAdaptorBase, 676
- set\_min\_sigma
  - mappel::Gauss1DsMAP, 232
  - mappel::Gauss1DsMLE, 260
  - mappel::Gauss1DsModel, 288
  - mappel::Gauss2DsMAP, 408
  - mappel::Gauss2DsMLE, 440
  - mappel::Gauss2DsModel, 471
  - mappel::Gauss2DsxyModel, 524
- set\_param\_names
  - mappel::Gauss1DMAP, 148
  - mappel::Gauss1DMLE, 176
  - mappel::Gauss1DModel, 203
  - mappel::Gauss1DsMAP, 232
  - mappel::Gauss1DsMLE, 260
  - mappel::Gauss1DsModel, 288
  - mappel::Gauss2DMAP, 317
  - mappel::Gauss2DMLE, 347
  - mappel::Gauss2DModel, 376
  - mappel::Gauss2DsMAP, 408
  - mappel::Gauss2DsMLE, 440
  - mappel::Gauss2DsModel, 472
  - mappel::Gauss2DsxyMAP, 499
  - mappel::Gauss2DsxyModel, 524
  - mappel::MAPEstimator, 588
  - mappel::MCMCAdaptor1Ds, 631
  - mappel::MCMCAdaptor1D, 614
  - mappel::MCMCAdaptor2Ds, 668
  - mappel::MCMCAdaptor2D, 649
  - mappel::MLEstimator, 694
  - mappel::PointEmitterModel, 756
- set\_prior
  - mappel::Gauss1DMAP, 149
  - mappel::Gauss1DMLE, 177
  - mappel::Gauss1DModel, 203
  - mappel::Gauss1DsMAP, 232
  - mappel::Gauss1DsMLE, 260
  - mappel::Gauss1DsModel, 288
  - mappel::Gauss2DMAP, 317
  - mappel::Gauss2DMLE, 347
  - mappel::Gauss2DModel, 376
  - mappel::Gauss2DsMAP, 408
  - mappel::Gauss2DsMLE, 440
  - mappel::Gauss2DsModel, 472
  - mappel::Gauss2DsxyMAP, 499
  - mappel::Gauss2DsxyModel, 524
  - mappel::MAPEstimator, 588
  - mappel::MCMCAdaptor1Ds, 632
  - mappel::MCMCAdaptor1D, 614
  - mappel::MCMCAdaptor2Ds, 668
  - mappel::MCMCAdaptor2D, 650
  - mappel::MLEstimator, 694
  - mappel::PointEmitterModel, 757
- set\_psf\_sigma

- mappel::Gauss1DModel, 204
- mappel::Gauss1DsMAP, 233
- mappel::Gauss1DsMLE, 261
- mappel::Gauss1DsModel, 289
- mappel::Gauss2DModel, 376
- mappel::Gauss2DsxyMAP, 500
- set\_rng\_seed
  - mappel::Gauss1DModel, 204
  - mappel::Gauss1DsMAP, 232
  - mappel::Gauss1DsMLE, 260
  - mappel::Gauss1DsModel, 288
  - mappel::Gauss2DModel, 377
  - mappel::Gauss2DsMAP, 408
  - mappel::Gauss2DsMLE, 441
  - mappel::Gauss2DsModel, 472
  - mappel::Gauss2DsxyMAP, 500
  - mappel::Gauss2DsxyModel, 524
  - mappel::MAPEstimator, 588
  - mappel::MCMCAdaptor1Ds, 632
  - mappel::MCMCAdaptor1D, 614
  - mappel::MCMCAdaptor2Ds, 669
  - mappel::MCMCAdaptor2D, 650
  - mappel::MLEstimator, 694
  - mappel::PointEmitterModel, 757
- set\_size
  - mappel::Gauss1DModel, 204
  - mappel::Gauss1DsMAP, 232, 233
  - mappel::Gauss1DsMLE, 261
  - mappel::Gauss1DsModel, 289
  - mappel::Gauss2DModel, 377
  - mappel::Gauss2DsMAP, 408
  - mappel::Gauss2DsMLE, 441
  - mappel::Gauss2DsModel, 472
  - mappel::Gauss2DsxyMAP, 500
  - mappel::Gauss2DsxyModel, 524
  - mappel::ImageFormat1DBase, 548
  - mappel::ImageFormat2DBase, 555
  - mappel::PoissonNoise1DObjective, 768
  - mappel::PoissonNoise2DObjective, 775
- set\_stencil
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, 597
- set\_ubound
  - mappel::Gauss1DModel, 204
  - mappel::Gauss1DsMLE, 260
- mappel::Gauss1DModel, 204
- mappel::Gauss1DsMAP, 233
- mappel::Gauss1DsMLE, 261
- mappel::Gauss1DsModel, 289
- mappel::Gauss2DModel, 377
- mappel::Gauss2DsMAP, 409
- mappel::Gauss2DsMLE, 441
- mappel::Gauss2DsModel, 472
- mappel::Gauss2DsxyMAP, 500
- mappel::Gauss2DsxyModel, 525
- mappel::MAPEstimator, 588
- mappel::MCMCAdaptor1Ds, 632
- mappel::MCMCAdaptor1D, 614
- mappel::MCMCAdaptor2Ds, 669
- mappel::MCMCAdaptor2D, 650
- mappel::MLEstimator, 694
- mappel::PointEmitterModel, 757
- setGlobalDefaultStrategy
  - omp\_exception\_catcher::impl\_::OMPEXception←Catcher, 744
- sgn
  - mappel, 44
- sigma
  - mappel::Gauss1DsModel::Stencil, 824
- sigma\_ratio
  - mappel::Gauss2DsModel::Stencil, 833
- sigma\_scale
  - mappel::Gauss1DModel, 210
  - mappel::Gauss1DsMAP, 238
  - mappel::Gauss1DsMLE, 267
  - mappel::Gauss1DsModel, 294
  - mappel::Gauss2DModel, 383
  - mappel::Gauss2DsMAP, 415
  - mappel::Gauss2DsMLE, 447
  - mappel::Gauss2DsModel, 478
  - mappel::Gauss2DsxyMAP, 506
  - mappel::MCMCAdaptor1Ds, 636
  - mappel::MCMCAdaptor1D, 618
  - mappel::MCMCAdaptor2Ds, 673
  - mappel::MCMCAdaptor2D, 654
  - mappel::MCMCAdaptorBase, 677
- sigmaX
  - mappel::Gauss2DsModel::Stencil, 833
  - mappel::Gauss2DsxyModel::Stencil, 819
- sigmaY
  - mappel::Gauss2DsModel::Stencil, 833
  - mappel::Gauss2DsxyModel::Stencil, 819
- simulate\_image

- mappel::methods, 64–66
  - PoissonGaussianNoise2DObjective.h, 932
- simulate\_image\_from\_model
  - mappel::methods, 66
- simulate\_image\_stack
  - mappel::methods::openmp, 89
- SimulatedAnnealingMaximizer
  - mappel::estimator::SimulatedAnnealingMaximizer, 807
- size
  - mappel::Gauss1DModel, 155
  - mappel::Gauss1DModel, 183
  - mappel::Gauss1DModel, 210
  - mappel::Gauss1DsMAP, 239
  - mappel::Gauss1DsMLE, 267
  - mappel::Gauss1DsModel, 294
  - mappel::Gauss2DModel, 324
  - mappel::Gauss2DModel, 354
  - mappel::Gauss2DModel, 383
  - mappel::Gauss2DsMAP, 415
  - mappel::Gauss2DsMLE, 447
  - mappel::Gauss2DsModel, 479
  - mappel::Gauss2DsxyMAP, 506
  - mappel::Gauss2DsxyModel, 529
  - mappel::ImageFormat1DBase, 548
  - mappel::ImageFormat2DBase, 555
  - mappel::PoissonNoise1DObjective, 769
  - mappel::PoissonNoise2DObjective, 776
- solve\_TR\_subproblem
  - mappel::estimator::subroutine, 53
- solve\_profile\_bound
  - mappel::estimator::IterativeMaximizer, 572
  - mappel::estimator::NewtonDiagonalMaximizer, 715
  - mappel::estimator::NewtonMaximizer, 736
  - mappel::estimator::QuasiNewtonMaximizer, 800
  - mappel::estimator::TrustRegionMaximizer, 870
- solve\_profile\_initial\_step
  - mappel::estimator::subroutine, 53
- solve\_restricted\_step\_length\_newton
  - mappel::estimator::subroutine, 53
- square
  - mappel, 44
- StatsT
  - mappel, 28
- Stencil
  - mappel::Gauss1DModel::Stencil, 838
  - mappel::Gauss1DsModel::Stencil, 823
  - mappel::Gauss2DModel::Stencil, 827
  - mappel::Gauss2DsModel::Stencil, 832
  - mappel::Gauss2DsxyModel::Stencil, 819
- stencil
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 597
- stencil.cpp, 939
- stencil.h, 940
- StencilVecT
  - mappel, 29
  - mappel::Gauss1DModel, 134
  - mappel::Gauss1DModel, 162
  - mappel::Gauss1DModel, 190
  - mappel::Gauss1DsMAP, 217
  - mappel::Gauss1DsMLE, 245
  - mappel::Gauss1DsModel, 274
  - mappel::Gauss2DModel, 302
  - mappel::Gauss2DModel, 332
  - mappel::Gauss2DModel, 361
  - mappel::Gauss2DsMAP, 391
  - mappel::Gauss2DsMLE, 423
  - mappel::Gauss2DsModel, 455
  - mappel::Gauss2DsxyMAP, 486
  - mappel::Gauss2DsxyModel, 512
- StencilT
  - mappel, 28
- step
  - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 601
- StepSize
  - mappel::estimator, 50
- Strategy
  - omp\_exception\_catcher, 91
- StringVecT
  - mappel, 29
- Success
  - mappel::estimator, 50
- TERM\_BLACK
  - mappel, 45
- TERM\_BLUE
  - mappel, 45
- TERM\_CYAN
  - mappel, 45
- TERM\_DIM\_BLACK
  - mappel, 45
- TERM\_DIM\_BLUE
  - mappel, 45
- TERM\_DIM\_CYAN
  - mappel, 45
- TERM\_DIM\_GREEN
  - mappel, 46
- TERM\_DIM\_MAGENTA
  - mappel, 46
- TERM\_DIM\_RED
  - mappel, 46
- TERM\_DIM\_WHITE
  - mappel, 46
- TERM\_DIM\_YELLOW
  - mappel, 46
- TERM\_GREEN

- mappel, 46
- TERM\_MAGENTA
  - mappel, 46
- TERM\_RED
  - mappel, 46
- TERM\_WHITE
  - mappel, 47
- TERM\_YELLOW
  - mappel, 47
- target\_rllh\_delta
  - mappel::estimator::ProfileBoundsData, 780
  - mappel::estimator::ProfileBoundsDataStack, 783
- theta
  - mappel::Gauss1DModel::Stencil, 840
  - mappel::Gauss1DsModel::Stencil, 826
  - mappel::Gauss2DModel::Stencil, 830
  - mappel::Gauss2DsModel::Stencil, 836
  - mappel::Gauss2DsxyModel::Stencil, 822
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, 598
- theta\_in\_bounds
  - mappel::Gauss1DModel, 150
  - mappel::Gauss1DModel, 178
  - mappel::Gauss1DModel, 204
  - mappel::Gauss1DsModel, 233
  - mappel::Gauss1DsModel, 261
  - mappel::Gauss1DsModel, 289
  - mappel::Gauss2DModel, 318
  - mappel::Gauss2DModel, 348
  - mappel::Gauss2DModel, 377
  - mappel::Gauss2DsModel, 409
  - mappel::Gauss2DsModel, 441
  - mappel::Gauss2DsModel, 472
  - mappel::Gauss2DsxyModel, 500
  - mappel::Gauss2DsxyModel, 525
  - mappel::MAPEstimator, 588
  - mappel::MCMCAdaptor1Ds, 632
  - mappel::MCMCAdaptor1Ds, 614
  - mappel::MCMCAdaptor2Ds, 669
  - mappel::MCMCAdaptor2D, 650
  - mappel::MLEstimator, 694
  - mappel::PointEmitterModel, 757
- theta\_seq
  - mappel::estimator::IterativeMaximizer::Maximizer←Data, 601
- theta\_stack\_in\_bounds
  - mappel::Gauss1DModel, 150
  - mappel::Gauss1DModel, 178
  - mappel::Gauss1DModel, 205
  - mappel::Gauss1DsModel, 233
  - mappel::Gauss1DsModel, 261
  - mappel::Gauss1DsModel, 289
  - mappel::Gauss2DModel, 318
  - mappel::Gauss2DModel, 348
- mappel::Gauss2DModel, 377
- mappel::Gauss2DsMAP, 409
- mappel::Gauss2DsMLE, 441
- mappel::Gauss2DsModel, 473
- mappel::Gauss2DsxyMAP, 501
- mappel::Gauss2DsxyModel, 525
- mappel::MAPEstimator, 589
- mappel::MCMCAdaptor1Ds, 632
- mappel::MCMCAdaptor1D, 615
- mappel::MCMCAdaptor2Ds, 669
- mappel::MCMCAdaptor2D, 650
- mappel::MLEstimator, 695
- mappel::PointEmitterModel, 757
- thin
  - mappel::mcmc::MCMCData, 679
  - mappel::mcmc::MCMCDataStack, 681
- thin\_sample
  - mappel::mcmc, 55
- ThreadedEstimator
  - mappel::estimator::ThreadedEstimator, 842
- total\_backtracks
  - mappel::estimator::IterativeMaximizer, 575
  - mappel::estimator::NewtonDiagonalMaximizer, 719
  - mappel::estimator::NewtonMaximizer, 740
  - mappel::estimator::QuasiNewtonMaximizer, 804
  - mappel::estimator::TrustRegionMaximizer, 875
- total\_der\_evals
  - mappel::estimator::IterativeMaximizer, 576
  - mappel::estimator::NewtonDiagonalMaximizer, 719
  - mappel::estimator::NewtonMaximizer, 740
  - mappel::estimator::QuasiNewtonMaximizer, 804
  - mappel::estimator::TrustRegionMaximizer, 875
- total\_fun\_evals
  - mappel::estimator::IterativeMaximizer, 576
  - mappel::estimator::NewtonDiagonalMaximizer, 719
  - mappel::estimator::NewtonMaximizer, 740
  - mappel::estimator::QuasiNewtonMaximizer, 804
  - mappel::estimator::TrustRegionMaximizer, 875
- total\_iterations
  - mappel::estimator::IterativeMaximizer, 576
  - mappel::estimator::NewtonDiagonalMaximizer, 719
  - mappel::estimator::NewtonMaximizer, 740
  - mappel::estimator::QuasiNewtonMaximizer, 804
  - mappel::estimator::TrustRegionMaximizer, 875
- total\_walltime
  - mappel::estimator::CGaussHeuristicEstimator, 104
  - mappel::estimator::CGaussMLE, 117
  - mappel::estimator::Estimator, 128
  - mappel::estimator::HeuristicEstimator, 542
  - mappel::estimator::IterativeMaximizer, 576
  - mappel::estimator::NewtonDiagonalMaximizer, 720
  - mappel::estimator::NewtonMaximizer, 741
  - mappel::estimator::QuasiNewtonMaximizer, 805



- mappel::estimator::SimulatedAnnealingMaximizer, 817
- mappel::estimator::ThreadedEstimator, 853
- mappel::estimator::TrustRegionMaximizer, 875
- trust\_radius\_decrease
  - mappel::estimator::TrustRegionMaximizer, 875
- trust\_radius\_decrease\_min
  - mappel::estimator::TrustRegionMaximizer, 876
- trust\_radius\_increase
  - mappel::estimator::TrustRegionMaximizer, 876
- TrustRegionMaximizer
  - mappel::estimator::TrustRegionMaximizer, 857
- TrustRegionRadius
  - mappel::estimator, 50
- ubound
  - mappel::Gauss1DModel, 156
  - mappel::Gauss1DModel, 184
  - mappel::Gauss1DModel, 210
  - mappel::Gauss1DsModel, 239
  - mappel::Gauss1DsModel, 267
  - mappel::Gauss1DsModel, 295
  - mappel::Gauss2DModel, 324
  - mappel::Gauss2DModel, 354
  - mappel::Gauss2DModel, 383
  - mappel::Gauss2DsModel, 416
  - mappel::Gauss2DsModel, 448
  - mappel::Gauss2DsModel, 479
  - mappel::Gauss2DsModel, 506
  - mappel::Gauss2DsModel, 530
  - mappel::MAPEstimator, 591
  - mappel::MCMCAdaptor1D, 637
  - mappel::MCMCAdaptor1D, 619
  - mappel::MCMCAdaptor2D, 674
  - mappel::MCMCAdaptor2D, 655
  - mappel::MLEstimator, 697
  - mappel::PointEmitterModel, 760
- Unassigned
  - mappel::estimator, 50
- UniformDistT
  - mappel, 29
- update\_internal\_1D\_estimators
  - mappel::Gauss2DsModel, 525
- update\_internal\_1Dsum\_estimators
  - mappel::Gauss2DModel, 318
  - mappel::Gauss2DModel, 348
  - mappel::Gauss2DModel, 377
  - mappel::Gauss2DsModel, 409
  - mappel::Gauss2DsModel, 441
  - mappel::Gauss2DsModel, 473
- util.cpp, 941
- util.h, 942
- VecFieldT
  - mappel, 29
- VecT
  - mappel, 29
- X
  - mappel::Gauss1DModel::Stencil, 840
  - mappel::Gauss1DsModel::Stencil, 826
  - mappel::Gauss2DModel::Stencil, 830
  - mappel::Gauss2DsModel::Stencil, 836
  - mappel::Gauss2DsxyModel::Stencil, 822
- x
  - mappel::Gauss1DModel::Stencil, 838
  - mappel::Gauss1DsModel::Stencil, 824
  - mappel::Gauss2DModel::Stencil, 828
  - mappel::Gauss2DsModel::Stencil, 833
  - mappel::Gauss2DsxyModel::Stencil, 819
- x\_model
  - mappel::Gauss2DModel, 325
  - mappel::Gauss2DModel, 355
  - mappel::Gauss2DModel, 384
  - mappel::Gauss2DsModel, 416
  - mappel::Gauss2DsModel, 448
  - mappel::Gauss2DsModel, 479
  - mappel::Gauss2DsxyModel, 530
- Y
  - mappel::Gauss2DModel::Stencil, 830
  - mappel::Gauss2DsModel::Stencil, 836
  - mappel::Gauss2DsxyModel::Stencil, 822
- y
  - mappel::Gauss2DModel::Stencil, 828
  - mappel::Gauss2DsModel::Stencil, 833
  - mappel::Gauss2DsxyModel::Stencil, 820
- y\_model
  - mappel::Gauss2DModel, 325
  - mappel::Gauss2DModel, 355
  - mappel::Gauss2DModel, 384
  - mappel::Gauss2DsModel, 416
  - mappel::Gauss2DsModel, 448
  - mappel::Gauss2DsModel, 480
  - mappel::Gauss2DsxyModel, 530