

Mappel

Generated by Doxygen 1.8.11

Contents

1	MAPPEL	2
2	Install	5
3	OMPExcceptionCatcher	8
4	Namespace Index	9
4.1	Namespace List	9
5	Hierarchical Index	10
5.1	Class Hierarchy	10
6	Class Index	13
6.1	Class List	13
7	File Index	16
7.1	File List	16
8	Namespace Documentation	20
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	49
8.2.3	Function Documentation	50
8.3	mappel::estimator::subroutine Namespace Reference	50
8.3.1	Detailed Description	51

8.3.2	Function Documentation	51
8.4	mappel::mcmc Namespace Reference	53
8.4.1	Function Documentation	54
8.5	mappel::methods Namespace Reference	55
8.5.1	Detailed Description	58
8.5.2	Function Documentation	58
8.6	mappel::methods::debug Namespace Reference	66
8.6.1	Function Documentation	67
8.7	mappel::methods::likelihood Namespace Reference	67
8.7.1	Function Documentation	68
8.8	mappel::methods::likelihood::debug Namespace Reference	70
8.8.1	Function Documentation	71
8.9	mappel::methods::objective Namespace Reference	72
8.9.1	Function Documentation	73
8.10	mappel::methods::objective::debug Namespace Reference	77
8.10.1	Function Documentation	78
8.11	mappel::methods::objective::openmp Namespace Reference	79
8.11.1	Function Documentation	80
8.12	mappel::methods::openmp Namespace Reference	84
8.12.1	Function Documentation	85
8.13	omp_exception_catcher Namespace Reference	89
8.13.1	Typedef Documentation	89
8.13.2	Enumeration Type Documentation	90
8.14	omp_exception_catcher::impl_ Namespace Reference	90

9	Class Documentation	90
9.1	mappel::ArrayShapeError Struct Reference	90
9.1.1	Detailed Description	90
9.1.2	Constructor & Destructor Documentation	91
9.2	mappel::ArraySizeError Struct Reference	91
9.2.1	Detailed Description	91
9.2.2	Constructor & Destructor Documentation	91
9.3	mappel::estimator::CGaussHeuristicEstimator< Model > Class Template Reference	92
9.3.1	Detailed Description	93
9.3.2	Constructor & Destructor Documentation	93
9.3.3	Member Function Documentation	93
9.3.4	Member Data Documentation	102
9.4	mappel::estimator::CGaussMLE< Model > Class Template Reference	104
9.4.1	Detailed Description	105
9.4.2	Constructor & Destructor Documentation	105
9.4.3	Member Function Documentation	106
9.4.4	Member Data Documentation	114
9.5	mappel::estimator::Estimator< Model > Class Template Reference	116
9.5.1	Detailed Description	118
9.5.2	Constructor & Destructor Documentation	118
9.5.3	Member Function Documentation	118
9.5.4	Friends And Related Function Documentation	126
9.5.5	Member Data Documentation	126
9.6	mappel::Gauss1DMAP Class Reference	127
9.6.1	Detailed Description	132
9.6.2	Member Typedef Documentation	132
9.6.3	Constructor & Destructor Documentation	134
9.6.4	Member Function Documentation	134

9.6.5	Member Data Documentation	149
9.7	mappel::Gauss1DMLE Class Reference	156
9.7.1	Detailed Description	161
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	185
9.8.1	Detailed Description	189
9.8.2	Member Typedef Documentation	190
9.8.3	Constructor & Destructor Documentation	191
9.8.4	Member Function Documentation	192
9.8.5	Member Data Documentation	206
9.9	mappel::Gauss1DsMAP Class Reference	213
9.9.1	Detailed Description	217
9.9.2	Member Typedef Documentation	217
9.9.3	Constructor & Destructor Documentation	219
9.9.4	Member Function Documentation	220
9.9.5	Member Data Documentation	236
9.10	mappel::Gauss1DsMLE Class Reference	242
9.10.1	Detailed Description	247
9.10.2	Member Typedef Documentation	247
9.10.3	Constructor & Destructor Documentation	249
9.10.4	Member Function Documentation	249
9.10.5	Member Data Documentation	265
9.11	mappel::Gauss1DsModel Class Reference	272
9.11.1	Detailed Description	276
9.11.2	Member Typedef Documentation	276

9.11.3	Constructor & Destructor Documentation	278
9.11.4	Member Function Documentation	278
9.11.5	Member Data Documentation	294
9.12	mappel::Gauss2DMAP Class Reference	300
9.12.1	Detailed Description	305
9.12.2	Member Typedef Documentation	305
9.12.3	Constructor & Destructor Documentation	307
9.12.4	Member Function Documentation	308
9.12.5	Member Data Documentation	324
9.13	mappel::Gauss2DMLE Class Reference	332
9.13.1	Detailed Description	336
9.13.2	Member Typedef Documentation	337
9.13.3	Constructor & Destructor Documentation	339
9.13.4	Member Function Documentation	339
9.13.5	Member Data Documentation	355
9.14	mappel::Gauss2DModel Class Reference	363
9.14.1	Detailed Description	368
9.14.2	Member Typedef Documentation	368
9.14.3	Constructor & Destructor Documentation	370
9.14.4	Member Function Documentation	370
9.14.5	Member Data Documentation	386
9.15	mappel::Gauss2DsMAP Class Reference	393
9.15.1	Detailed Description	398
9.15.2	Member Typedef Documentation	398
9.15.3	Constructor & Destructor Documentation	400
9.15.4	Member Function Documentation	401
9.15.5	Member Data Documentation	419
9.16	mappel::Gauss2DsMLE Class Reference	427

9.16.1	Detailed Description	432
9.16.2	Member Typedef Documentation	432
9.16.3	Constructor & Destructor Documentation	434
9.16.4	Member Function Documentation	435
9.16.5	Member Data Documentation	452
9.17	mappel::Gauss2DsModel Class Reference	460
9.17.1	Detailed Description	465
9.17.2	Member Typedef Documentation	465
9.17.3	Constructor & Destructor Documentation	467
9.17.4	Member Function Documentation	468
9.17.5	Member Data Documentation	484
9.18	mappel::Gauss2DsxyMAP Class Reference	492
9.18.1	Detailed Description	496
9.18.2	Member Typedef Documentation	496
9.18.3	Constructor & Destructor Documentation	498
9.18.4	Member Function Documentation	498
9.18.5	Member Data Documentation	513
9.19	mappel::Gauss2DsxyModel Class Reference	520
9.19.1	Detailed Description	524
9.19.2	Member Typedef Documentation	524
9.19.3	Constructor & Destructor Documentation	526
9.19.4	Member Function Documentation	526
9.19.5	Member Data Documentation	538
9.20	mappel::estimator::HeuristicEstimator< Model > Class Template Reference	544
9.20.1	Detailed Description	545
9.20.2	Constructor & Destructor Documentation	546
9.20.3	Member Function Documentation	546
9.20.4	Member Data Documentation	555

9.21	mappel::ImageFormat1DBase Class Reference	557
9.21.1	Detailed Description	558
9.21.2	Member Typedef Documentation	558
9.21.3	Constructor & Destructor Documentation	560
9.21.4	Member Function Documentation	560
9.21.5	Member Data Documentation	562
9.22	mappel::ImageFormat2DBase Class Reference	563
9.22.1	Detailed Description	564
9.22.2	Member Typedef Documentation	564
9.22.3	Constructor & Destructor Documentation	566
9.22.4	Member Function Documentation	566
9.22.5	Member Data Documentation	568
9.23	mappel::estimator::IterativeMaximizer< Model > Class Template Reference	569
9.23.1	Detailed Description	572
9.23.2	Constructor & Destructor Documentation	572
9.23.3	Member Function Documentation	572
9.23.4	Member Data Documentation	585
9.24	mappel::LogicalError Struct Reference	590
9.24.1	Detailed Description	590
9.24.2	Constructor & Destructor Documentation	590
9.25	mappel::MAPEstimator Class Reference	590
9.25.1	Detailed Description	593
9.25.2	Member Typedef Documentation	593
9.25.3	Constructor & Destructor Documentation	593
9.25.4	Member Function Documentation	594
9.25.5	Member Data Documentation	603
9.26	mappel::estimator::IterativeMaximizer< Model >::MaximizerData Class Reference	606
9.26.1	Detailed Description	608

9.26.2	Constructor & Destructor Documentation	608
9.26.3	Member Function Documentation	609
9.26.4	Member Data Documentation	612
9.27	mappel::MCMCAdaptor1D Class Reference	615
9.27.1	Detailed Description	618
9.27.2	Member Typedef Documentation	618
9.27.3	Constructor & Destructor Documentation	619
9.27.4	Member Function Documentation	619
9.27.5	Member Data Documentation	629
9.28	mappel::MCMCAdaptor1Ds Class Reference	634
9.28.1	Detailed Description	637
9.28.2	Member Typedef Documentation	637
9.28.3	Constructor & Destructor Documentation	638
9.28.4	Member Function Documentation	638
9.28.5	Member Data Documentation	648
9.29	mappel::MCMCAdaptor2D Class Reference	654
9.29.1	Detailed Description	657
9.29.2	Member Typedef Documentation	657
9.29.3	Constructor & Destructor Documentation	657
9.29.4	Member Function Documentation	658
9.29.5	Member Data Documentation	668
9.30	mappel::MCMCAdaptor2Ds Class Reference	673
9.30.1	Detailed Description	676
9.30.2	Member Typedef Documentation	676
9.30.3	Constructor & Destructor Documentation	677
9.30.4	Member Function Documentation	677
9.30.5	Member Data Documentation	687
9.31	mappel::MCMCAdaptorBase Class Reference	693

9.31.1 Detailed Description	694
9.31.2 Constructor & Destructor Documentation	694
9.31.3 Member Function Documentation	694
9.31.4 Member Data Documentation	695
9.32 mappel::mcmc::MCMCData Struct Reference	696
9.32.1 Detailed Description	696
9.32.2 Member Function Documentation	696
9.32.3 Member Data Documentation	696
9.33 mappel::mcmc::MCMCDataStack Struct Reference	698
9.33.1 Detailed Description	698
9.33.2 Member Function Documentation	699
9.33.3 Member Data Documentation	699
9.34 mappel::mcmc::MCMCDebugData Struct Reference	700
9.34.1 Detailed Description	700
9.34.2 Member Function Documentation	701
9.34.3 Member Data Documentation	701
9.35 mappel::estimator::MLEDebugData Struct Reference	702
9.35.1 Detailed Description	702
9.35.2 Member Function Documentation	702
9.35.3 Member Data Documentation	702
9.36 mappel::MLEstimator Class Reference	704
9.36.1 Detailed Description	707
9.36.2 Member Typedef Documentation	707
9.36.3 Constructor & Destructor Documentation	707
9.36.4 Member Function Documentation	708
9.36.5 Member Data Documentation	717
9.37 mappel::ModelBoundsError Struct Reference	720
9.37.1 Detailed Description	721

9.37.2	Constructor & Destructor Documentation	721
9.38	mappel::estimator::NewtonDiagonalMaximizer< Model > Class Template Reference	721
9.38.1	Detailed Description	724
9.38.2	Member Typedef Documentation	724
9.38.3	Constructor & Destructor Documentation	724
9.38.4	Member Function Documentation	725
9.38.5	Member Data Documentation	738
9.39	mappel::estimator::NewtonMaximizer< Model > Class Template Reference	742
9.39.1	Detailed Description	745
9.39.2	Member Typedef Documentation	745
9.39.3	Constructor & Destructor Documentation	745
9.39.4	Member Function Documentation	746
9.39.5	Member Data Documentation	759
9.40	mappel::NotImplementedError Struct Reference	763
9.40.1	Detailed Description	763
9.40.2	Constructor & Destructor Documentation	764
9.41	mappel::NumericalError Struct Reference	764
9.41.1	Detailed Description	764
9.41.2	Constructor & Destructor Documentation	764
9.42	omp_exception_catcher::impl::OMPExcceptionCatcher< _dummy > Class Template Reference	765
9.42.1	Detailed Description	765
9.42.2	Constructor & Destructor Documentation	765
9.42.3	Member Function Documentation	766
9.43	mappel::ParameterValueError Struct Reference	767
9.43.1	Detailed Description	767
9.43.2	Constructor & Destructor Documentation	767
9.44	mappel::PointEmitterModel Class Reference	768
9.44.1	Detailed Description	771

9.44.2	Member Typedef Documentation	771
9.44.3	Constructor & Destructor Documentation	771
9.44.4	Member Function Documentation	772
9.44.5	Member Data Documentation	780
9.45	PoissonGaussianNoise2DObjective< ModelBase > Class Template Reference	784
9.45.1	Detailed Description	784
9.45.2	Member Typedef Documentation	785
9.45.3	Constructor & Destructor Documentation	785
9.45.4	Member Data Documentation	785
9.46	mappel::PoissonNoise1DObjective Class Reference	786
9.46.1	Detailed Description	788
9.46.2	Member Typedef Documentation	788
9.46.3	Constructor & Destructor Documentation	789
9.46.4	Member Function Documentation	790
9.46.5	Member Data Documentation	792
9.47	mappel::PoissonNoise2DObjective Class Reference	793
9.47.1	Detailed Description	795
9.47.2	Member Typedef Documentation	795
9.47.3	Constructor & Destructor Documentation	797
9.47.4	Member Function Documentation	797
9.47.5	Member Data Documentation	799
9.48	mappel::estimator::ProfileBoundsData Struct Reference	801
9.48.1	Detailed Description	801
9.48.2	Member Function Documentation	801
9.48.3	Member Data Documentation	802
9.49	mappel::estimator::ProfileBoundsDataStack Struct Reference	804
9.49.1	Detailed Description	805
9.49.2	Member Function Documentation	805

9.49.3	Member Data Documentation	805
9.50	mappel::estimator::QuasiNewtonMaximizer< Model > Class Template Reference	807
9.50.1	Detailed Description	810
9.50.2	Member Typedef Documentation	810
9.50.3	Constructor & Destructor Documentation	810
9.50.4	Member Function Documentation	811
9.50.5	Member Data Documentation	824
9.51	mappel::estimator::SimulatedAnnealingMaximizer< Model > Class Template Reference	828
9.51.1	Detailed Description	830
9.51.2	Constructor & Destructor Documentation	830
9.51.3	Member Function Documentation	830
9.51.4	Member Data Documentation	838
9.52	mappel::Gauss2DsxyModel::Stencil Class Reference	840
9.52.1	Detailed Description	841
9.52.2	Member Typedef Documentation	841
9.52.3	Constructor & Destructor Documentation	842
9.52.4	Member Function Documentation	842
9.52.5	Friends And Related Function Documentation	843
9.52.6	Member Data Documentation	843
9.53	mappel::Gauss1DsModel::Stencil Class Reference	845
9.53.1	Detailed Description	846
9.53.2	Member Typedef Documentation	846
9.53.3	Constructor & Destructor Documentation	846
9.53.4	Member Function Documentation	847
9.53.5	Friends And Related Function Documentation	848
9.53.6	Member Data Documentation	848
9.54	mappel::Gauss2DModel::Stencil Class Reference	849
9.54.1	Detailed Description	850

9.54.2	Member Typedef Documentation	850
9.54.3	Constructor & Destructor Documentation	850
9.54.4	Member Function Documentation	851
9.54.5	Friends And Related Function Documentation	852
9.54.6	Member Data Documentation	852
9.55	mappel::Gauss2DsModel::Stencil Class Reference	854
9.55.1	Detailed Description	855
9.55.2	Member Typedef Documentation	855
9.55.3	Constructor & Destructor Documentation	855
9.55.4	Member Function Documentation	855
9.55.5	Friends And Related Function Documentation	857
9.55.6	Member Data Documentation	857
9.56	mappel::Gauss1DModel::Stencil Class Reference	859
9.56.1	Detailed Description	860
9.56.2	Member Typedef Documentation	860
9.56.3	Constructor & Destructor Documentation	861
9.56.4	Member Function Documentation	861
9.56.5	Friends And Related Function Documentation	862
9.56.6	Member Data Documentation	862
9.57	mappel::estimator::ThreadedEstimator< Model > Class Template Reference	863
9.57.1	Detailed Description	865
9.57.2	Constructor & Destructor Documentation	865
9.57.3	Member Function Documentation	865
9.57.4	Member Data Documentation	874
9.58	mappel::estimator::TrustRegionMaximizer< Model > Class Template Reference	876
9.58.1	Detailed Description	879
9.58.2	Member Typedef Documentation	879
9.58.3	Constructor & Destructor Documentation	880
9.58.4	Member Function Documentation	880
9.58.5	Member Data Documentation	894

10 File Documentation	899
10.1 display.cpp File Reference	899
10.1.1 Detailed Description	900
10.2 display.h File Reference	900
10.2.1 Detailed Description	901
10.3 estimator.cpp File Reference	901
10.3.1 Detailed Description	902
10.4 estimator.h File Reference	902
10.4.1 Detailed Description	903
10.4.2 Class Documentation	903
10.4.3 Typedef Documentation	905
10.5 estimator_helpers.h File Reference	905
10.5.1 Detailed Description	906
10.6 estimator_impl.h File Reference	906
10.6.1 Detailed Description	906
10.7 estimator_statics.cpp File Reference	907
10.7.1 Detailed Description	907
10.8 Gauss1DMAP.cpp File Reference	907
10.8.1 Detailed Description	907
10.9 Gauss1DMAP.h File Reference	908
10.9.1 Detailed Description	908
10.10Gauss1DMLE.cpp File Reference	908
10.10.1 Detailed Description	909
10.11Gauss1DMLE.h File Reference	909
10.11.1 Detailed Description	909
10.12Gauss1DModel.cpp File Reference	910
10.12.1 Detailed Description	910
10.13Gauss1DModel.h File Reference	910

10.13.1 Detailed Description	911
10.14Gauss1DsMAP.cpp File Reference	911
10.14.1 Detailed Description	911
10.15Gauss1DsMAP.h File Reference	911
10.15.1 Detailed Description	912
10.16Gauss1DsMLE.cpp File Reference	912
10.16.1 Detailed Description	912
10.17Gauss1DsMLE.h File Reference	913
10.17.1 Detailed Description	913
10.18Gauss1DsModel.cpp File Reference	913
10.18.1 Detailed Description	914
10.19Gauss1DsModel.h File Reference	914
10.19.1 Detailed Description	914
10.20Gauss2DMAP.cpp File Reference	915
10.20.1 Detailed Description	915
10.21Gauss2DMAP.h File Reference	915
10.21.1 Detailed Description	916
10.22Gauss2DMLE.cpp File Reference	916
10.22.1 Detailed Description	916
10.23Gauss2DMLE.h File Reference	916
10.23.1 Detailed Description	917
10.24Gauss2DModel.cpp File Reference	917
10.24.1 Detailed Description	917
10.25Gauss2DModel.h File Reference	918
10.25.1 Detailed Description	918
10.26Gauss2DsMAP.cpp File Reference	919
10.26.1 Detailed Description	919
10.27Gauss2DsMAP.h File Reference	919

10.27.1 Detailed Description	920
10.28Gauss2DsMLE.cpp File Reference	920
10.28.1 Detailed Description	920
10.29Gauss2DsMLE.h File Reference	920
10.29.1 Detailed Description	921
10.30Gauss2DsModel.cpp File Reference	921
10.30.1 Detailed Description	921
10.31Gauss2DsModel.h File Reference	922
10.31.1 Detailed Description	922
10.32Gauss2DsxyMAP.h File Reference	923
10.32.1 Detailed Description	923
10.33Gauss2DsxyModel.h File Reference	923
10.33.1 Detailed Description	924
10.34ImageFormat1DBase.cpp File Reference	924
10.34.1 Detailed Description	925
10.35ImageFormat1DBase.h File Reference	925
10.35.1 Detailed Description	925
10.36ImageFormat2DBase.cpp File Reference	926
10.36.1 Detailed Description	926
10.37ImageFormat2DBase.h File Reference	926
10.37.1 Detailed Description	927
10.38Install.md File Reference	927
10.39MAPEstimator.h File Reference	927
10.39.1 Detailed Description	928
10.40mcmc.cpp File Reference	928
10.40.1 Detailed Description	929
10.41mcmc.h File Reference	929
10.41.1 Detailed Description	930

10.42mcmc_data.h File Reference	930
10.42.1 Detailed Description	931
10.43MCMCAdaptor1D.cpp File Reference	931
10.43.1 Detailed Description	931
10.44MCMCAdaptor1D.h File Reference	931
10.44.1 Detailed Description	932
10.45MCMCAdaptor1Ds.cpp File Reference	932
10.45.1 Detailed Description	932
10.46MCMCAdaptor1Ds.h File Reference	933
10.46.1 Detailed Description	933
10.47MCMCAdaptor2D.cpp File Reference	933
10.47.1 Detailed Description	933
10.48MCMCAdaptor2D.h File Reference	934
10.48.1 Detailed Description	934
10.49MCMCAdaptor2Ds.cpp File Reference	934
10.49.1 Detailed Description	934
10.50MCMCAdaptor2Ds.h File Reference	935
10.50.1 Detailed Description	935
10.51MCMCAdaptorBase.cpp File Reference	935
10.51.1 Detailed Description	935
10.52MCMCAdaptorBase.h File Reference	936
10.52.1 Detailed Description	936
10.53MLEstimator.h File Reference	936
10.53.1 Detailed Description	937
10.54model_methods.h File Reference	938
10.55model_methods_impl.h File Reference	941
10.56numerical.cpp File Reference	944
10.56.1 Detailed Description	945

10.57numerical.h File Reference	945
10.57.1 Detailed Description	946
10.58OMPExcptionCatcher.h File Reference	946
10.58.1 Detailed Description	947
10.59openmp_methods.h File Reference	947
10.59.1 Detailed Description	949
10.60PointEmitterModel.cpp File Reference	950
10.60.1 Detailed Description	950
10.61PointEmitterModel.h File Reference	951
10.61.1 Detailed Description	951
10.62PoissonGaussianNoise2DObjective.cpp File Reference	952
10.63PoissonGaussianNoise2DObjective.h File Reference	952
10.63.1 Detailed Description	953
10.63.2 Function Documentation	953
10.64PoissonNoise1DObjective.cpp File Reference	955
10.64.1 Detailed Description	955
10.65PoissonNoise1DObjective.h File Reference	955
10.65.1 Detailed Description	957
10.66PoissonNoise2DObjective.cpp File Reference	957
10.66.1 Detailed Description	957
10.67PoissonNoise2DObjective.h File Reference	957
10.67.1 Detailed Description	959
10.68README.md File Reference	959
10.69README.md File Reference	959
10.70rng.cpp File Reference	959
10.70.1 Detailed Description	960
10.71rng.h File Reference	960
10.71.1 Detailed Description	961
10.72stencil.cpp File Reference	961
10.72.1 Detailed Description	962
10.73stencil.h File Reference	962
10.73.1 Detailed Description	964
10.74util.cpp File Reference	964
10.75util.h File Reference	964
10.75.1 Detailed Description	966

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](#) fluorescence 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

- Copyright: 2013-2019
- Author: Mark J. Olah
- Email: (`mjo@cs.unm DOT 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 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:

mappel	20
mappel::estimator	47
mappel::estimator::subroutine	50
mappel::mcmc	53
mappel::methods	
Templated functions for operating on a PointEmitterModel	55
mappel::methods::debug	66
mappel::methods::likelihood	67

mappel::methods::likelihood::debug	70
mappel::methods::objective	72
mappel::methods::objective::debug	77
mappel::methods::objective::openmp	79
mappel::methods::openmp	84
omp_exception_catcher	89
omp_exception_catcher::impl_	90

5 Hierarchical Index

5.1 Class Hierarchy

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

mappel::estimator::Estimator< Model >	116
mappel::estimator::ThreadedEstimator< Model >	863
mappel::estimator::CGaussHeuristicEstimator< Model >	92
mappel::estimator::CGaussMLE< Model >	104
mappel::estimator::HeuristicEstimator< Model >	544
mappel::estimator::IterativeMaximizer< Model >	569
mappel::estimator::NewtonDiagonalMaximizer< Model >	721
mappel::estimator::NewtonMaximizer< Model >	742
mappel::estimator::QuasiNewtonMaximizer< Model >	807
mappel::estimator::TrustRegionMaximizer< Model >	876
mappel::estimator::SimulatedAnnealingMaximizer< Model >	828
mappel::ImageFormat1DBase	557
mappel::Gauss1DModel	185
mappel::Gauss1DMAP	127
mappel::Gauss1DMLE	156
mappel::Gauss2DsxyMAP	492
mappel::Gauss1DsModel	272
mappel::Gauss1DsMAP	213

mappel::Gauss1DsMLE	242
mappel::PoissonNoise1DObjective	786
mappel::Gauss1DMAP	127
mappel::Gauss1DMLE	156
mappel::Gauss1DsMAP	213
mappel::Gauss1DsMLE	242
mappel::Gauss2DsxyMAP	492
ImageFormat1DBase	
PoissonGaussianNoise2DObjective< ModelBase >	784
mappel::ImageFormat2DBase	563
mappel::Gauss2DModel	363
mappel::Gauss2DMAP	300
mappel::Gauss2DMLE	332
mappel::Gauss2DsModel	460
mappel::Gauss2DsMAP	393
mappel::Gauss2DsMLE	427
mappel::Gauss2DsxyModel	520
mappel::PoissonNoise2DObjective	793
mappel::Gauss2DMAP	300
mappel::Gauss2DMLE	332
mappel::Gauss2DsMAP	393
mappel::Gauss2DsMLE	427
MappelError	
mappel::ArrayShapeError	90
mappel::ArraySizeError	91
mappel::LogicalError	590
mappel::ModelBoundsError	720
mappel::NotImplementedError	763
mappel::NumericalError	764
mappel::ParameterValueError	767

mappel::estimator::IterativeMaximizer< Model >::MaximizerData	606
mappel::MCMCAdaptorBase	693
mappel::MCMCAdaptor1D	615
mappel::Gauss1DModel	185
mappel::MCMCAdaptor1Ds	634
mappel::Gauss1DsModel	272
mappel::MCMCAdaptor2D	654
mappel::Gauss2DModel	363
mappel::MCMCAdaptor2Ds	673
mappel::Gauss2DsModel	460
mappel::mcmc::MCMCData	696
mappel::mcmc::MCMCDataStack	698
mappel::mcmc::MCMCDebugData	700
mappel::estimator::MLEData	47
mappel::estimator::MLEDataStack	47
mappel::estimator::MLEDebugData	702
omp_exception_catcher::impl_::OMPExcptionCatcher< _dummy >	765
mappel::PointEmitterModel	768
mappel::Gauss1DModel	185
mappel::Gauss1DsModel	272
mappel::Gauss2DModel	363
mappel::Gauss2DsModel	460
mappel::Gauss2DsxyModel	520
mappel::MAPEstimator	590
mappel::Gauss1DMAP	127
mappel::Gauss1DsMAP	213
mappel::Gauss2DMAP	300
mappel::Gauss2DsMAP	393
mappel::Gauss2DsxyMAP	492
mappel::MCMCAdaptor1D	615

mappel::MLEstimator	704
mappel::Gauss1DMLE	156
mappel::Gauss1DsMLE	242
mappel::Gauss2DMLE	332
mappel::Gauss2DsMLE	427
mappel::estimator::ProfileBoundsData	801
mappel::estimator::ProfileBoundsDataStack	804
mappel::estimator::ProfileBoundsDebugData	47
mappel::estimator::ProfileLikelihoodData	47
mappel::Gauss2DsxyModel::Stencil	840
mappel::Gauss1DsModel::Stencil	845
mappel::Gauss2DModel::Stencil	849
mappel::Gauss2DsModel::Stencil	854
mappel::Gauss1DModel::Stencil	859

6 Class Index

6.1 Class List

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

mappel::ArrayShapeError	
Array is not of the right dimensionality	90
mappel::ArraySizeError	
Array is not of the right size	91
mappel::estimator::CGaussHeuristicEstimator< Model >	92
mappel::estimator::CGaussMLE< Model >	104
mappel::estimator::Estimator< Model >	116
mappel::Gauss1DMAP	
A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	127
mappel::Gauss1DMLE	
A 1D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective	156
mappel::Gauss1DModel	
A base class for 1D Gaussian PSF with a fixed sigma (standard dev.)	185

<code>mappel::Gauss1DsMAP</code>	
A 1D Gaussian with variable PSF sigma under an Poisson read noise assumption and MAP Objective	213
<code>mappel::Gauss1DsMLE</code>	
A 1D Gaussian with variable PSF under an Poisson noise assumption and maximum-likelihood estimator	242
<code>mappel::Gauss1DsModel</code>	
Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels	272
<code>mappel::Gauss2DMAP</code>	
A 2D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	300
<code>mappel::Gauss2DMLE</code>	
A 2D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective	332
<code>mappel::Gauss2DModel</code>	
A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma	363
<code>mappel::Gauss2DsMAP</code>	
A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum a-posteriori objective	393
<code>mappel::Gauss2DsMLE</code>	
A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum-likelihood objective	427
<code>mappel::Gauss2DsModel</code>	
A base class for 2D Gaussian PSF where the gaussian sigma is controlled by a single scalar parameter which is called <code>sigma_ratio</code> . The size of the gaussian psf is <code>sigma_ratio*psf_sigma</code> , where <code>psf_sigma</code> is considered as a vector [<code>psf_sigmaX</code> , <code>psf_sigmaY</code>]	460
<code>mappel::Gauss2DsxyMAP</code>	
A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	492
<code>mappel::Gauss2DsxyModel</code>	
A base class for 2D Gaussian PSF with axis-aligned gaussian with free parameters for both <code>sigma_x</code> and <code>sigma_y</code> . Gaussian sigma parameters <code>sigma_x</code> and <code>sigma_y</code> are measured in units of pixels. The model has 6 parameters, [<code>x</code> , <code>y</code> , <code>l</code> , <code>bg</code> , <code>sigma_x</code> , <code>sigma_y</code>]	520
<code>mappel::estimator::HeuristicEstimator< Model ></code>	544
<code>mappel::ImageFormat1DBase</code>	
A virtual base class for 2D image localization objectives	557
<code>mappel::ImageFormat2DBase</code>	
A virtual base class for 2D image localization objectives	563
<code>mappel::estimator::IterativeMaximizer< Model ></code>	569
<code>mappel::LogicalError</code>	
Failure of code or algorithm logic	590

mappel::MAPEstimator	
A Mixin class to configure a for MLE estimation (null prior)	590
mappel::estimator::IterativeMaximizer< Model >::MaximizerData	606
mappel::MCMCAdaptor1D	615
mappel::MCMCAdaptor1Ds	634
mappel::MCMCAdaptor2D	654
mappel::MCMCAdaptor2Ds	673
mappel::MCMCAdaptorBase	693
mappel::mcmc::MCMCData	696
mappel::mcmc::MCMCDataStack	698
mappel::mcmc::MCMCDebugData	700
mappel::estimator::MLEDebugData	702
mappel::MLEstimator	
A Mixin class to configure a for MLE estimation (null prior)	704
mappel::ModelBoundsError	
Access outside the model bounds is attempted	720
mappel::estimator::NewtonDiagonalMaximizer< Model >	721
mappel::estimator::NewtonMaximizer< Model >	742
mappel::NotImplementedError	
Feature not yet implemented	763
mappel::NumericalError	
Expected numerical condition does not hold	764
omp_exception_catcher::impl_::OMPExcceptionCatcher< _dummy >	765
mappel::ParameterValueError	
Parameter value is not valid	767
mappel::PointEmitterModel	
A virtual Base type for point emitter localization models	768
PoissonGaussianNoise2DObjective< ModelBase >	
A Base type for point emitter localization models that use 2d images	784
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	786

<code>mappel::PoissonNoise2DObjective</code>	
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	793
<code>mappel::estimator::ProfileBoundsData</code>	801
<code>mappel::estimator::ProfileBoundsDataStack</code>	804
<code>mappel::estimator::QuasiNewtonMaximizer< Model ></code>	807
<code>mappel::estimator::SimulatedAnnealingMaximizer< Model ></code>	828
<code>mappel::Gauss2DsxyModel::Stencil</code>	
Stencil for 2D free-sigma (astigmatic) models	840
<code>mappel::Gauss1DsModel::Stencil</code>	
Stencil for 1D variable-sigma models	845
<code>mappel::Gauss2DModel::Stencil</code>	
Stencil for 2D fixed-sigma models	849
<code>mappel::Gauss2DsModel::Stencil</code>	
Stencil for 2D scalar-sigma models	854
<code>mappel::Gauss1DModel::Stencil</code>	
Stencil for 1D fixed-sigma models	859
<code>mappel::estimator::ThreadedEstimator< Model ></code>	863
<code>mappel::estimator::TrustRegionMaximizer< Model ></code>	876

7 File Index

7.1 File List

Here is a list of all files with brief descriptions:

<code>display.cpp</code>	899
<code>display.h</code>	
Textual image display with colors	900
<code>estimator.cpp</code>	
Non-templated estimator helper routines and static constants	901
<code>estimator.h</code>	
The class declaration and inline and templated functions for the Estimator class hierarchy	902
<code>estimator_helpers.h</code>	
Estimator helper subroutines	905
<code>estimator_impl.h</code>	906

estimator_statics.cpp	907
Gauss1DMAP.cpp	
The class definition and template Specializations for Gauss1DMAP	907
Gauss1DMAP.h	
The class declaration and inline and templated functions for Gauss1DMAP	908
Gauss1DMLE.cpp	
The class definition and template Specializations for Gauss1DMLE	908
Gauss1DMLE.h	
The class declaration and inline and templated functions for Gauss1DMLE	909
Gauss1DModel.cpp	
The class definition and template Specializations for Gauss1DModel	910
Gauss1DModel.h	
The class declaration and inline and templated functions for Gauss1DModel	910
Gauss1DsMAP.cpp	
The class definition and template Specializations for Gauss1DsMAP	911
Gauss1DsMAP.h	
The class declaration and inline and templated functions for Gauss1DsMAP	911
Gauss1DsMLE.cpp	
The class definition and template Specializations for Gauss1DsMLE	912
Gauss1DsMLE.h	
The class declaration and inline and templated functions for Gauss1DsMLE	913
Gauss1DsModel.cpp	
The class definition and template Specializations for Gauss1DsModel	913
Gauss1DsModel.h	
The class declaration and inline and templated functions for Gauss1DsModel	914
Gauss2DMAP.cpp	
The class definition and template Specializations for Gauss2DMAP	915
Gauss2DMAP.h	
The class declaration and inline and templated functions for Gauss2DMAP	915
Gauss2DMLE.cpp	
The class definition and template Specializations for Gauss2DMLE	916
Gauss2DMLE.h	
The class declaration and inline and templated functions for Gauss2DMLE	916
Gauss2DModel.cpp	
The class definition and template Specializations for Gauss2DModel	917
Gauss2DModel.h	
The class declaration and inline and templated functions for Gauss2DModel	918

Gauss2DsMAP.cpp	
The class definition and template Specializations for Gauss2DsMAP	919
Gauss2DsMAP.h	
The class declaration and inline and templated functions for Gauss2DsMAP	919
Gauss2DsMLE.cpp	
The class definition and template Specializations for Gauss2DsMLE	920
Gauss2DsMLE.h	
The class declaration and inline and templated functions for Gauss2DsMLE	920
Gauss2DsModel.cpp	
The class definition and template Specializations for Gauss2DsModel	921
Gauss2DsModel.h	
The class declaration and inline and templated functions for Gauss2DsModel	922
Gauss2DsxyMAP.h	
The class declaration and inline and templated functions for Gauss2DsxyMAP	923
Gauss2DsxyModel.h	
The class declaration and inline and templated functions for Gauss2DsxyModel	923
ImageFormat1DBase.cpp	
The class definition and template Specializations for ImageFormat1DBase	924
ImageFormat1DBase.h	
The class declaration and inline and templated functions for ImageFormat1DBase	925
ImageFormat2DBase.cpp	
The class definition and template Specializations for ImageFormat2DBase	926
ImageFormat2DBase.h	
The class declaration and inline and templated functions for ImageFormat2DBase	926
MAPEstimator.h	
Class declaration and inline and templated functions for MAPEstimator	927
mcmc.cpp	
MCMC helper functions	928
mcmc.h	
Templated MCMC methods for posterior estimation	929
mcmc_data.h	
MCMC data storage types	930
MCMCAdaptor1D.cpp	
The class definition and template Specializations for MCMCAdaptor1D	931
MCMCAdaptor1D.h	
The class declaration and inline and templated functions for MCMCAdaptor1D	931
MCMCAdaptor1Ds.cpp	
The class definition and template Specializations for MCMCAdaptor1Ds	932

MCMCAdaptor1Ds.h	
The class declaration and inline and templated functions for MCMCAdaptor1Ds	933
MCMCAdaptor2D.cpp	
The class definition and template Specializations for MCMCAdaptor2D	933
MCMCAdaptor2D.h	
The class declaration and inline and templated functions for MCMCAdaptor2D	934
MCMCAdaptor2Ds.cpp	
The class definition and template Specializations for MCMCAdaptor2Ds	934
MCMCAdaptor2Ds.h	
The class declaration and inline and templated functions for MCMCAdaptor2Ds	935
MCMCAdaptorBase.cpp	
The class definition and template Specializations for MCMCAdaptorBase	935
MCMCAdaptorBase.h	
The class declaration and inline and templated functions for MCMCAdaptorBase	936
MLEstimator.h	
Class declaration and inline and templated functions for MLEstimator	936
model_methods.h	938
model_methods_impl.h	941
numerical.cpp	
Numerical matrix operations	944
numerical.h	
Numerical matrix operations	945
OMPExcceptionCatcher.h	
A lightweight class for managing C++ exception handling strategies for OpenMP methods	946
openmp_methods.h	
Namespaces for OpenMP parallelized versions of the mappel::model namespace functions (external methods)	947
PointEmitterModel.cpp	
The class definition and template Specializations for PointEmitterModel	950
PointEmitterModel.h	
The class declaration and inline and templated functions for PointEmitterModel	951
PoissonGaussianNoise2DObjective.cpp	952
PoissonGaussianNoise2DObjective.h	
The class declaration and inline and templated functions for PoissonGaussianNoise2DObjective	952
PoissonNoise1DObjective.cpp	
The class definition and template Specializations for PoissonNoise1DObjective	955
PoissonNoise1DObjective.h	
The class declaration and inline and templated functions for PoissonNoise1DObjective	955

PoissonNoise2DObjective.cpp	
The class definition and template Specializations for PoissonNoise2DObjective	957
PoissonNoise2DObjective.h	
The class declaration and inline and templated functions for PoissonNoise2DObjective	957
rng.cpp	
Global random number generator	959
rng.h	
Random number generation usign sfmt	960
stencil.cpp	
The stencils for pixel based computations	961
stencil.h	
The stencils for pixel based computations	962
util.cpp	964
util.h	
Common utilities and errors	964

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 $Cx = b$.

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 283 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

MatT	obsI	Observed Fisher information matrix at theta.
double	rllh	RLLH at theta.
VecT	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 65 of file estimator.h.

Class Members

IdxT	Ndata	Number of data estimates.
CubeT	obsI	Observed Fisher information matrix stack. size:[Nparams,Nparams,Ndata].
VecT	rllh	RLLH stack. size:[Ndata].
MatT	theta	Theta estimate stack. size:[Nparams,Ndata].

8.2.1.3 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 (ouputut) parameters to give output parameters context.

Definition at line 115 of file estimator.h.

Class Members

IdxT	estimated_idx	Index of single parameter to estimate for.
MLEData	mle	Theta maximum-likelihood estimate, rllh, and Obsl.
IdxT	Nseq_lb	Number of points in sequence_lb.
IdxT	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.
MatT	sequence_lb	size:[NumParams,Nseq_lb] Sequence of evaluated points for lb estimate (including theta mle as initial point)
VecT	sequence_lb_rllh	size:[Nseq_lb] RLLH at each of the sequence_lb points
MatT	sequence_ub	size:[NumParams,Nseq_ub] Sequence of evaluated points for ub estimate (including theta mle as initial point)
VecT	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 ($-\chi^2_{\text{inv}}(\text{confidence},1)/2$)

8.2.1.4 struct mappel::estimator::ProfileLikelihoodData

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

Definition at line 76 of file estimator.h.

Class Members

IdxVecT	fixed_idxxs	Indexes of fixed parameters.
MatT	fixed_values	Vector values for each fixed parameter size:[Nfixed,Nvalues];.
IdxT	Nfixed	Number of fixed parameters.
IdxT	Nvalues	Number of values of fixed parameters evaluated.
VecT	profile_likelihood	profile likelihood for each column of fixed parameter values
MatT	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 174 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 91 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 142 of file estimator.cpp.

References mappel::sgn(), and mappel::estimator::MLEDebugData::theta.

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 185 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. sign(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 169 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 180 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 193 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 174 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 61 of file estimator.cpp.

References mappel::estimator::MLEDebugData::obsI.

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 266 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 Nocetal and Wright (2000) More and Sorensen (1981)

Definition at line 199 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 &rlh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`void aposteriori_objective` (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, double &rlh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`void prior_objective` (const Model &model, const ParamT< Model > &theta, double &rlh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`void likelihood_objective` (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, double &rlh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`void likelihood_objective` (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, double &rlh, 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 & rllh, ParamT< Model > & grad, MatT & hess)`

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

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

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 θ). Note: Expected information is an average quantity and is independent of the data. Enabled for [PoissonNoise1DObjective](#).

Parameters

<i>model</i>	PointEmitterModel
<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>	PolImageCoordTEmitterModel
<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 & rllh, 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

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

<i>in</i>	<i>model</i>	Model object
<i>in</i>	<i>model_im</i>	An image representing the expected (mean) at each pixel under the PSF model.
<i>in, out</i>	<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::Iterative↵Maximizer< 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 PointEmitterModel
--------------	--

Parameters

in	<i>model</i>	A PointEmitterModel 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 515 of file `openmp_methods.h`.

References `mappel::methods::objective::grad()`, `omp_exception_catcher::impl_::OMPExceptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl_::OMPExceptionCatcher<_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 PointEmitterModel
--------------	--

Parameters

in	<i>model</i>	A PointEmitterModel 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 563 of file `openmp_methods.h`.

References `mappel::methods::objective::hessian()`, `omp_exception_catcher::impl::OMPExcptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl::OMPExcptionCatcher<_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 PointEmitterModel
--------------	--

Parameters

in	<i>model</i>	A PointEmitterModel 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 379 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 PointEmitterModel
--------------	--

Parameters

in	<i>model</i>	A PointEmitterModel 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 404 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 PointEmitterModel
--------------	--

Parameters

in	<i>model</i>	A PointEmitterModel object.
in	<i>image_stack</i>	Sequence of images.

Parameters

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 612 of file openmp_methods.h.

References `mappel::methods::objective::negative_definite_hessian()`, `omp_exception_catcher::impl::OMPEXceptionCatcher<_dummy>::rethrow()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher<_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 PointEmitterModel
--------------	--

Parameters

in	<i>model</i>	A PointEmitterModel 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 451 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 487 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, estimator::MLEDataStack &mle_data_stack, StatsT &stats)`
- `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 273 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 294 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 323 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`, `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 333 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 342 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 352 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, estimator::MLEDataStack &
mle_data_stack, StatsT & stats)`

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)`

Definition at line 189 of file openmp_methods.h.

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

8.12.1.11 `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 198 of file `openmp_methods.h`.

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

8.12.1.12 `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 227 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::OMPExcceptionCatcher< _dummy >::rethrow()`, `omp_exception_catcher::impl::OMPExcceptionCatcher< _dummy >::run()`, and `mappel::mcmc::MCMCDataStack::thin`.

Referenced by `estimate_posterior_stack()`.

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

Definition at line 265 of file `openmp_methods.h`.

References `estimate_posterior_stack()`.

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)`

Definition at line 208 of file `openmp_methods.h`.

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

8.12.1.15 `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 217 of file `openmp_methods.h`.

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

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

8.12.1.17 `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 PointEmitterModel
--------------	--

Parameters

in	<i>model</i>	A PointEmitterModel 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_::OMPEXceptionCatcher< _dummy >::rethrow()`, and `omp_exception_catcher::impl_::OMPEXceptionCatcher< _dummy >::run()`.

8.12.1.18 `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 PointEmitterModel
--------------	--

Parameters

in	<i>model,A</i>	PointEmitterModel 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_::OMPEXceptionCatcher< _dummy >::rethrow()`, and `omp_exception_catcher::impl_::OMPEXceptionCatcher< _dummy >::run()`.

8.12.1.19 `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 PointEmitterModel
--------------	--

Parameters

in	<i>model</i>	A PointEmitterModel 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_::OMPExcceptionCatcher<_dummy>::rethrow()`, `omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::run()`, and `mappel::methods::simulate_image()`.

8.13 omp_exception_catcher Namespace Reference

Namespaces

- [impl_](#)

Typedefs

- using [OMPExcceptionCatcher](#) = [impl_::OMPExcceptionCatcher<>](#)

Enumerations

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

8.13.1 Typedef Documentation

8.13.1.1 using omp_exception_catcher::OMPExcceptionCatcher = typedef impl_::OMPExcceptionCatcher<>

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

Definition at line 114 of file OMPExcceptionCatcher.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](#)

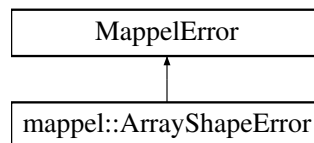
9 Class Documentation

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:

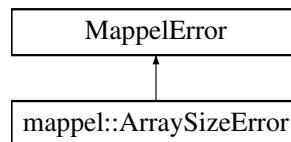
- [util.h](#)

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.

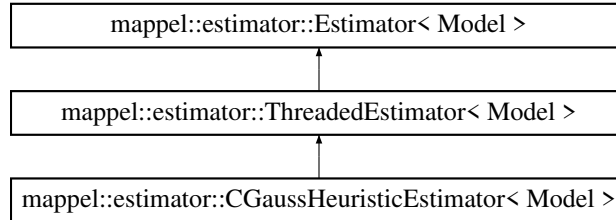
The documentation for this struct was generated from the following file:

- [util.h](#)

9.3 mappel::estimator::CGaussHeuristicEstimator< Model > 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, [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](#)

9.3.1 Detailed Description

```
template<class Model>
class mappel::estimator::CGaussHeuristicEstimator< Model >
```

Definition at line 348 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 351 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_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()`.

9.3.3.6 `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>	MLEData 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>	MLEData 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>	MLEData 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>	MLEDebugData 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>	MLEData 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>	MLEData 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`, `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`, `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().

9.3.3.17 `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 276 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 355 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 301 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 326 of file estimator.h.

9.3.4.3 `template<class Model > const Model& mappel::estimator::Estimator< Model >::model` [protected],[inherited]

Definition at line 296 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 328 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 299 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 327 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 300 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()`.

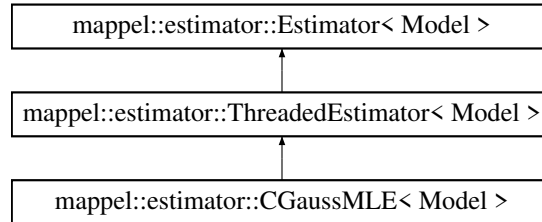
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

9.4 mappel::estimator::CGaussMLE< Model > 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 363 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 368 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>	MLEData 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>	MLEData 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>	MLEData 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>	MLEDebugData 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>	MLEData 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>	MLEData 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 276 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 373 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 366 of file `estimator.h`.

9.4.4.2 `template<class Model> IdxVecT mappel::estimator::Estimator< Model>::exit_counts` `[protected], [inherited]`

Definition at line 301 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 326 of file estimator.h.

9.4.4.4 `template<class Model> const Model& mappel::estimator::Estimator< Model >::model` [protected],
[inherited]

Definition at line 296 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 328 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 299 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 327 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 300 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()`.

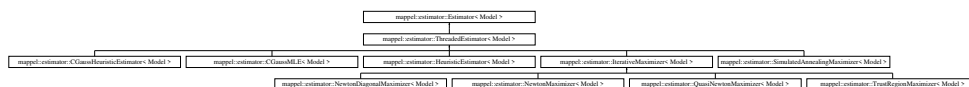
The documentation for this class was generated from the following files:

- [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 196 of file estimator.h.

9.5.2 Constructor & 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 199 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>	MLEData 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>	MLEData 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>	MLEData 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>	MLEDebugData 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>	MLEData 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>	MLEData 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_idxes, 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 276 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 301 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 296 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 299 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 300 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:

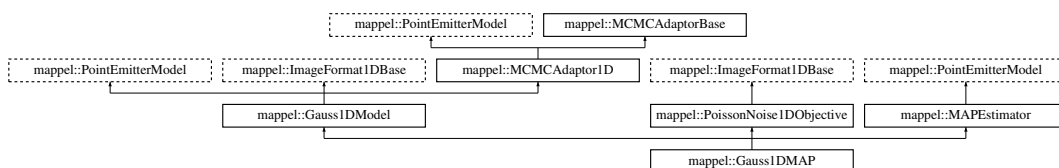
- [estimator.h](#)
- [estimator_impl.h](#)

9.6 mappel::Gauss1DMAP 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/Gauss1DMAP.h>
```

Inheritance diagram for `mappel::Gauss1DMAP`:



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=`DefaultPriorSigmaPos`)
- static `prior_hessian::ScaledSymmetricBetaDist make_prior_component_position_beta` (`IdxT` size, double pos_↵ beta=`DefaultPriorBetaPos`)
- static `prior_hessian::TruncatedGammaDist make_prior_component_intensity` (double mean=`DefaultPriorMeanI`, double kappa=`DefaultPriorIntensityKappa`)
- static `prior_hessian::TruncatedParetoDist make_prior_component_sigma` (double min_sigma, double max_↵ sigma, double alpha=`DefaultPriorPSFSigmaAlpha`)
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested].
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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](#)

9.6.1 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 & 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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.6.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline],[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.6.4.23 CompositeDist & mappel::PointEmitterModel::get_prior () [inline], [inherited]

Definition at line 207 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 211 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 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.6.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () `[static],[inherited]`

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanI, double kappa = DefaultPriorIntensityKappa)` `[static]`, `[inherited]`

Definition at line 105 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::DefaultPriorMaxI`.

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 = DefaultPriorBetaPos)` `[static]`, `[inherited]`

Definition at line 99 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()`.

9.6.4.53 **prior_hessian::TruncatedNormalDist** **mappel::PointEmitterModel::make_prior_component_position_normal** (*IdxT size*, *double pos_sigma = DefaultPriorSigmaPos*) [static],[inherited]

Definition at line 92 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()`.

9.6.4.54 **prior_hessian::TruncatedParetoDist** **mappel::PointEmitterModel::make_prior_component_sigma** (*double min_sigma*, *double max_sigma*, *double alpha = DefaultPriorPSFSigmaAlpha*) [static],[inherited]

Definition at line 111 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 283 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 323 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 251 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_u`, `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 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.6.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` [inline], [inherited]

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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 227 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::DefaultPriorMeanI, 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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.6.4.82 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 165 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()`.

9.6.4.83 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 156 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::Point←EmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.6.4.84 `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 117 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 244 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 264 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().

9.6.4.91 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const [inherited]`

Definition at line 303 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 67 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::DefaultConfidenceLevel = 0.95` [static],[inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.6.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` [static],[inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.6.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static],[inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.6.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.6.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.6.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.6.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.6.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.6.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

9.6.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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().

9.6.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.6.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.6.5.14 `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 53 of file Gauss1DModel.h.

Referenced by mappel::Gauss1DModel::operator=().

9.6.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.6.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],
[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.6.5.17 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names [static], [inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

9.6.5.18 `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()`.

9.6.5.19 `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()`.

9.6.5.20 `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()`.

9.6.5.21 `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()`.

9.6.5.22 `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()`.

9.6.5.23 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.6.5.24 `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()`.

9.6.5.25 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.6.5.26 `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()`.

9.6.5.27 `ParamT mappel::PointEmitterModel::lbound` [protected], [inherited]

Definition at line 155 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()`.

9.6.5.28 `const std::string mappel::Gauss1DModel::name` `[static]`

Definition at line 34 of file Gauss1DModel.h.

9.6.5.29 `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()`.

9.6.5.30 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.6.5.31 `IdxT mappel::PointEmitterModel::num_params` `[protected], [inherited]`

Definition at line 153 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()`.

9.6.5.32 `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()`.

9.6.5.33 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 152 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().

9.6.5.34 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=().

9.6.5.35 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().

9.6.5.36 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().

9.6.5.37 `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()`.

9.6.5.38 `ParamT mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 155 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:

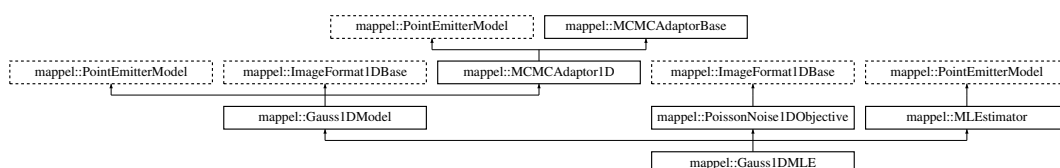
- [Gauss1DMAP.h](#)
- [Gauss1DMAP.cpp](#)

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=`DefaultPriorSigmaPos`)
- static `prior_hessian::ScaledSymmetricBetaDist make_prior_component_position_beta` (`IdxT` size, double pos_↔ beta=`DefaultPriorBetaPos`)
- static `prior_hessian::TruncatedGammaDist make_prior_component_intensity` (double mean=`DefaultPriorMeanI`, double kappa=`DefaultPriorIntensityKappa`)
- static `prior_hessian::TruncatedParetoDist make_prior_component_sigma` (double min_sigma, double max_↔ sigma, double alpha=`DefaultPriorPSFSigmaAlpha`)
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested].
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.7.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`, `[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.7.4.23 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline]`,`[inherited]`

Definition at line 207 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 211 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 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.7.4.28 **ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()** `[static],[inherited]`

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanI, double kappa = DefaultPriorIntensityKappa) [static], [inherited]`

Definition at line 105 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::DefaultPriorMaxI`.

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 = DefaultPriorBetaPos) [static], [inherited]`

Definition at line 99 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()`.

9.7.4.53 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal (IdxT size, double pos_sigma = DefaultPriorSigmaPos)` [static],[inherited]

Definition at line 92 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()`.

9.7.4.54 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma (double min_sigma, double max_sigma, double alpha = DefaultPriorPSFSigmaAlpha)` [static],[inherited]

Definition at line 111 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

<code>theta</code>	Parameter to evaluate at
<code>compute_derivatives</code>	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 283 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 323 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 251 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_u`, `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 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.7.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` [inline],[inherited]

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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 227 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::DefaultPriorMeanI, 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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.7.4.82 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 165 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()`.

9.7.4.83 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 156 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::Point←EmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.7.4.84 `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 117 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 244 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 264 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()`.

9.7.4.91 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const [inherited]`

Definition at line 303 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 67 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::DefaultConfidenceLevel = 0.95` `[static], [inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.7.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` `[static], [inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.7.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` `[static], [inherited]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.7.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` `[static], [inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.7.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` `[static], [inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.7.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.7.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.7.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.7.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.7.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.7.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static],[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.7.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.7.5.14 `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal"` `[static],[inherited]`

Definition at line 53 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::operator=()`.

9.7.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static],[inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.7.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],
[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.7.5.17 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names [static], [inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

9.7.5.18 `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()`.

9.7.5.19 `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()`.

9.7.5.20 `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()`.

9.7.5.21 `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()`.

9.7.5.22 `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()`.

9.7.5.23 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.7.5.24 `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()`.

9.7.5.25 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.7.5.26 `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()`.

9.7.5.27 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 155 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()`.

9.7.5.28 `const std::string mappel::Gauss1DMLE::name` `[static]`

Definition at line 34 of file Gauss1DMLE.h.

9.7.5.29 `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()`.

9.7.5.30 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.7.5.31 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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()`.

9.7.5.32 `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()`.

9.7.5.33 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 152 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().

9.7.5.34 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=().

9.7.5.35 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().

9.7.5.36 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().

9.7.5.37 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().

9.7.5.38 ParamT mappel::PointEmitterModel::ubound [protected], [inherited]

Definition at line 155 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:

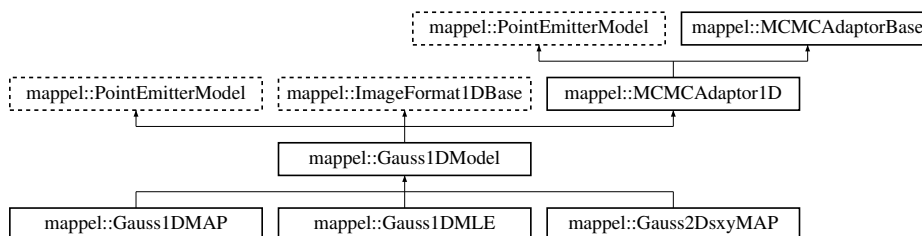
- [Gauss1DMLE.h](#)
- [Gauss1DMLE.cpp](#)

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=DefaultPriorSigmaPos)
- static prior_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta` (`IdxT` size, double pos_↵ beta=DefaultPriorBetaPos)
- static prior_hessian::TruncatedGammaDist `make_prior_component_intensity` (double mean=DefaultPriorMeanI, double kappa=DefaultPriorIntensityKappa)
- static prior_hessian::TruncatedParetoDist `make_prior_component_sigma` (double min_sigma, double max_↵ sigma, double alpha=DefaultPriorPSFSigmaAlpha)
- 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 `DefaultEstimatorMethod` = "TrustRegion"
- Default optimization method for MLE/MAP estimation.*
- static const std::string `DefaultProfileBoundsEstimatorMethod` = "Newton"
- Default optimization method for profile bounds optimizations.*
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
 - static const `IdxT` `DefaultMCMCNumSamples` = 300
- Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)*
- static const `IdxT` `DefaultMCMCBurnin` = 10
- Number of samples to throw away (burn-in) on initialization.*
- static const `IdxT` `DefaultMCMCThin` = 0
- Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]*

- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.8.4.12 StringVecT mappel::PointEmitterModel::get_hyperparam_names () const [inline],[inherited]

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.8.4.23 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline],[inherited]`

Definition at line 207 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 211 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 127 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.8.4.28 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]`

Definition at line 122 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 223 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxxyModel::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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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* = DefaultPriorMeanI, double *kappa* = DefaultPriorIntensityKappa) [static], [inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::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::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss2DsModel::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 = DefaultPriorBetaPos) [static],[inherited]`

Definition at line 99 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 = DefaultPriorSigmaPos) [static],[inherited]`

Definition at line 92 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 = DefaultPriorPSFSigmaAlpha) [static],[inherited]`

Definition at line 111 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 283 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 323 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 251 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_u`, `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 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.8.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` [inline],[inherited]

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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 227 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::DefaultPriorMeanI, 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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.8.4.82 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 165 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()`.

9.8.4.83 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 156 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::Point←EmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.8.4.84 `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 117 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 244 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 264 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::Gauss2DsModel::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 303 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 67 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::DefaultConfidenceLevel = 0.95` [static],[inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.8.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` [static],[inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.8.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static],[inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.8.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.8.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.8.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.8.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.8.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::make_prior_component_intensity().

9.8.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.8.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 64 of file PointEmitterModel.h.

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_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.8.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static],[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.8.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.8.5.14 `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal"` `[static]`

Definition at line 53 of file Gauss1DModel.h.

Referenced by `operator=()`.

9.8.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static],[inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.8.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],
[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.8.5.17 `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()`.

9.8.5.18 `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()`.

9.8.5.19 `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()`.

9.8.5.20 `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()`.

9.8.5.21 `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()`.

9.8.5.22 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.8.5.23 `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()`.

9.8.5.24 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.8.5.25 `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()`.

9.8.5.26 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 155 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()`.

9.8.5.27 `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().

9.8.5.28 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get_num_hyperparams(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set_prior().

9.8.5.29 `IdxT mappel::PointEmitterModel::num_params` `[protected], [inherited]`

Definition at line 153 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().

9.8.5.30 `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().

9.8.5.31 `CompositeDist mappel::PointEmitterModel::prior` `[protected], [inherited]`

Definition at line 152 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().

9.8.5.32 `const StringVecT mappel::Gauss1DModel::prior_types` `[static]`

Initial value:

```
= { "Beta",
                                     "Normal"
}
```

Definition at line 52 of file Gauss1DModel.h.

Referenced by operator=().

9.8.5.33 `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().

9.8.5.34 `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().

9.8.5.35 `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().

9.8.5.36 ParamT mappel::PointEmitterModel::ubound [protected], [inherited]

Definition at line 155 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:

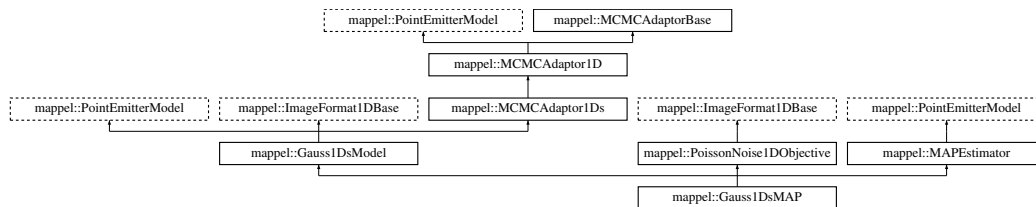
- [Gauss1DModel.h](#)
- [Gauss1DModel.cpp](#)

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=[DefaultPriorSigmaPos](#))
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_↵ beta=[DefaultPriorBetaPos](#))
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[DefaultPriorMeanI](#), double kappa=[DefaultPriorIntensityKappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵ sigma, double alpha=[DefaultPriorPSFSigmaAlpha](#))
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2

- 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 [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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.9.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const [inline],[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.9.4.25 CompositeDist & mappel::PointEmitterModel::get_prior () [inline], [inherited]

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.9.4.27 ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static], [inherited]

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.9.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static], [inherited]

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanI*, *double kappa = DefaultPriorIntensityKappa*) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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(), 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 = DefaultPriorBetaPos*) [static],[inherited]

Definition at line 99 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 = DefaultPriorSigmaPos*) [static],[inherited]

Definition at line 92 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* = DefaultPriorPSFSigmaAlpha) [static],[inherited]

Definition at line 111 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::l(), 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::l(), 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::l(), 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::l(), 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::l(), and mappel::Gauss1DsModel::Stencil::X.

9.9.4.64 **PointEmitterModel::ParamT** mappel::PointEmitterModel::reflected_theta (**const ParamT & theta**) **const**
[inherited]

Definition at line 283 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 323 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 251 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 271 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.9.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline]`, `[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG, 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 220 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 267 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 247 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 227 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::DefaultPriorMeanI`, `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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.9.4.86 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior)` [inherited]

Definition at line 165 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()`.

9.9.4.87 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior)` [inherited]

Definition at line 156 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::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 117 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 244 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 264 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().

9.9.4.93 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const [inherited]`

Definition at line 303 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 67 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()`.

9.9.5.2 `const double mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95` `[static], [inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

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

9.9.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` `[static], [inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file `PointEmitterModel.h`.

9.9.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` `[static], [inherited]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file `PointEmitterModel.h`.

9.9.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` `[static], [inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file `PointEmitterModel.h`.

9.9.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` `[static], [inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file `PointEmitterModel.h`.

9.9.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file `PointEmitterModel.h`.

9.9.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static], [inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.9.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static], [inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.9.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` [static], [inherited]

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.9.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` [static], [inherited]

Default per-pixel mean background counts

Definition at line 64 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()`.

9.9.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.9.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.9.5.14 `const std::string mappel::Gauss1DsModel::DefaultPriorType = "Normal" [static], [inherited]`

Definition at line 51 of file Gauss1DsModel.h.

Referenced by `mappel::Gauss1DsModel::operator=()`.

9.9.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton" [static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.9.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.9.5.17 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names [static], [inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

9.9.5.18 `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()`.

9.9.5.19 `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()`.

9.9.5.20 `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()`.

9.9.5.21 `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()`.

9.9.5.22 `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()`.

9.9.5.23 `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()`.

9.9.5.24 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.9.5.25 `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()`.

9.9.5.26 `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 68 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.9.5.27 `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()`.

9.9.5.28 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 155 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()`.

9.9.5.29 `const std::string mappel::Gauss1DsMAP::name` `[static]`

Definition at line 35 of file `Gauss1DsMAP.h`.

9.9.5.30 `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()`.

9.9.5.31 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

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

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.9.5.32 `IdxT mappel::PointEmitterModel::num_params` [protected],[inherited]

Definition at line 153 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()`.

9.9.5.33 `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()`.

9.9.5.34 `CompositeDist mappel::PointEmitterModel::prior` [protected],[inherited]

Definition at line 152 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()`.

9.9.5.35 `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=()`.

9.9.5.36 `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()`.

9.9.5.37 `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()`.

9.9.5.38 `ParamT mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 155 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:

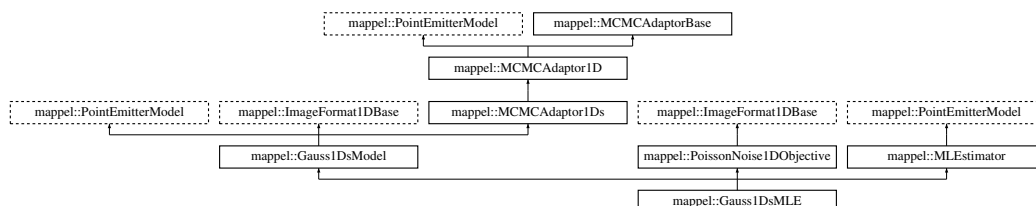
- [Gauss1DsMAP.h](#)
- [Gauss1DsMAP.cpp](#)

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=[DefaultPriorSigmaPos](#))
- static [prior_hessian::ScaledSymmetricBetaDist make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↵ beta=[DefaultPriorBetaPos](#))
- static [prior_hessian::TruncatedGammaDist make_prior_component_intensity](#) (double mean=[DefaultPriorMeanI](#), double kappa=[DefaultPriorIntensityKappa](#))
- static [prior_hessian::TruncatedParetoDist make_prior_component_sigma](#) (double min_sigma, double max_↵ sigma, double alpha=[DefaultPriorPSFSigmaAlpha](#))
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.10.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`, `[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.10.4.25 **CompositeDist &** mappel::PointEmitterModel::get_prior () [inline],[inherited]

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.10.4.27 **ParallelRngGeneratorT &** mappel::PointEmitterModel::get_rng_generator () [static],[inherited]

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.10.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanl, double kappa = DefaultPriorIntensityKappa)` `[static], [inherited]`

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxl.

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().

9.10.4.52 **prior_hessian::ScaledSymmetricBetaDist** **mappel::PointEmitterModel::make_prior_component_position_beta** (*IdxT size*, *double pos_beta = DefaultPriorBetaPos*) [static],[inherited]

Definition at line 99 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 = DefaultPriorSigmaPos*) [static],[inherited]

Definition at line 92 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()**.

9.10.4.54 **prior_hessian::TruncatedParetoDist** **mappel::PointEmitterModel::make_prior_component_sigma** (*double min_sigma*, *double max_sigma*, *double alpha = DefaultPriorPSFSigmaAlpha*) [static],[inherited]

Definition at line 111 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 283 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 323 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 251 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 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.10.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` [inline],[inherited]

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitter`
`Model::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.10.4.73 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc) [inline],
[inherited]`

Definition at line 267 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 247 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 227 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::DefaultPriorMeanI, 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 233 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 259 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.10.4.86 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_) [inherited]`

Definition at line 165 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 156 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 117 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 244 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 264 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()`.

9.10.4.93 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const` [inherited]

Definition at line 303 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 67 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()`.

9.10.5.2 `const double mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95` [static], [inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.10.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` [static], [inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.10.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static], [inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.10.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.10.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested].

Definition at line 56 of file PointEmitterModel.h.

9.10.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.10.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.10.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.10.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` [static],[inherited]

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.10.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` [static],[inherited]

Default per-pixel mean background counts

Definition at line 64 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()`.

9.10.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.10.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.10.5.14 `const std::string mappel::Gauss1DsModel::DefaultPriorType = "Normal"` [static], [inherited]

Definition at line 51 of file Gauss1DsModel.h.

Referenced by `mappel::Gauss1DsModel::operator=()`.

9.10.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` [static], [inherited]

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.10.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` [static], [inherited]

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.10.5.17 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names` [static], [inherited]

Definition at line 24 of file PoissonNoise1DObjective.h.

9.10.5.18 `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()`.

9.10.5.19 `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()`.

9.10.5.20 `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()`.

9.10.5.21 `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()`.

9.10.5.22 `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()`.

9.10.5.23 `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()`.

9.10.5.24 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.10.5.25 `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()`.

9.10.5.26 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.10.5.27 `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()`.

9.10.5.28 `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 155 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()`.

9.10.5.29 `const std::string mappel::Gauss1DsMLE::name` `[static]`

Definition at line 35 of file Gauss1DsMLE.h.

9.10.5.30 `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()`.

9.10.5.31 **IdxT** `mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

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

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.10.5.32 **IdxT** `mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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()`.

9.10.5.33 **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()`.

9.10.5.34 **CompositeDist** `mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 152 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()`.

9.10.5.35 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=().

9.10.5.36 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().

9.10.5.37 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().

9.10.5.38 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 155 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:

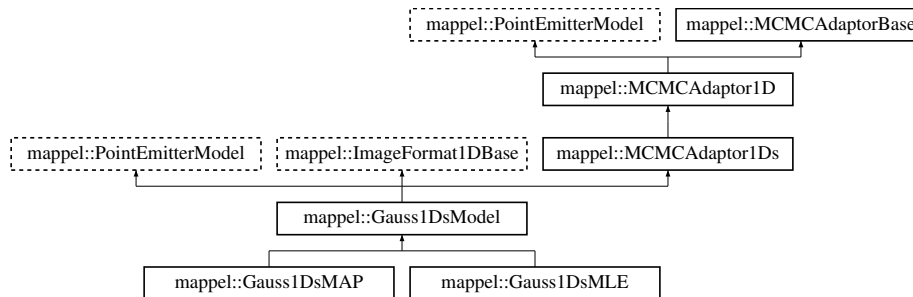
- [Gauss1DsMLE.h](#)
- [Gauss1DsMLE.cpp](#)

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=[DefaultPriorSigmaPos](#))
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_↵ beta=[DefaultPriorBetaPos](#))
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[DefaultPriorMeanI](#), double kappa=[DefaultPriorIntensityKappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵ sigma, double alpha=[DefaultPriorPSFSigmaAlpha](#))
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const IdxT [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const IdxT [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const IdxT [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.11.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`, `[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.11.4.25 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline],[inherited]`

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.11.4.27 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static],[inherited]`

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.11.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanl, double kappa = DefaultPriorIntensityKappa)` `[static]`, `[inherited]`

Definition at line 105 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::DefaultPriorMaxl`.

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 = DefaultPriorBetaPos)` `[static]`, `[inherited]`

Definition at line 99 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 = DefaultPriorSigmaPos*) [static], [inherited]

Definition at line 92 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 = DefaultPriorPSFSigmaAlpha*) [static], [inherited]

Definition at line 111 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::I(), 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::I(), 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::I(), 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::I()`, `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::I()`, and `mappel::Gauss1DsModel::Stencil::X`.

9.11.4.64 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const [inherited]`

Definition at line 283 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 323 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 251 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_I`, `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 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.11.4.70 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` [inline],[inherited]

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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 227 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::DefaultPriorMeanI, 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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.11.4.86 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 165 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 156 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 117 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 244 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 264 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::Gauss2DsModel::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 303 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 67 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().

9.11.5.2 **const double** mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95 [static],[inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.11.5.3 **const std::string** mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion" [static],[inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.11.5.4 **const IdxT** mappel::PointEmitterModel::DefaultMCMCBurnin = 10 [static],[inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.11.5.5 **const IdxT** mappel::PointEmitterModel::DefaultMCMCNumSamples = 300 [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.11.5.6 **const IdxT** mappel::PointEmitterModel::DefaultMCMCThin = 0 [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.11.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.11.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.11.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.11.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.11.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.11.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.11.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.11.5.14 `const std::string mappel::Gauss1DsModel::DefaultPriorType = "Normal" [static]`

Definition at line 51 of file Gauss1DsModel.h.

Referenced by operator=().

9.11.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton" [static],
[inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.11.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],
[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), and mappel::Gauss2DsModel::initial_theta_estimate().

9.11.5.17 `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().

9.11.5.18 `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().

9.11.5.19 `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().

9.11.5.20 `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()`.

9.11.5.21 `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()`.

9.11.5.22 `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()`.

9.11.5.23 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.11.5.24 `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()`.

9.11.5.25 `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 68 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.11.5.26 `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()`.

9.11.5.27 `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 155 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()`.

9.11.5.28 `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()`.

9.11.5.29 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.11.5.30 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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()`.

9.11.5.31 `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()`.

9.11.5.32 `CompositeDist mappel::PointEmitterModel::prior` [protected],[inherited]

Definition at line 152 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()`.

9.11.5.33 `const StringVecT mappel::Gauss1DsModel::prior_types` [static]

Initial value:

```
= { "Beta",
    "Normal"
}
```

Definition at line 50 of file Gauss1DsModel.h.

Referenced by `operator=()`.

9.11.5.34 `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()`.

9.11.5.35 `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()`.

9.11.5.36 `ParamT` `mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 155 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:

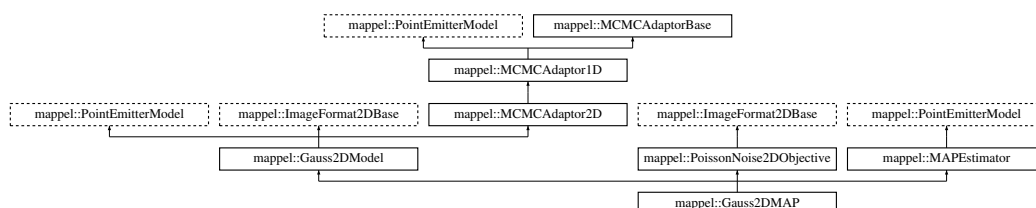
- [Gauss1DsModel.h](#)
- [Gauss1DsModel.cpp](#)

9.12 `mappel::Gauss2DMap` 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/Gauss2DMap.h>
```

Inheritance diagram for `mappel::Gauss2DMap`:



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=[DefaultPriorSigmaPos](#))
 - static [prior_hessian::ScaledSymmetricBetaDist make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↔ beta=[DefaultPriorBetaPos](#))
 - static [prior_hessian::TruncatedGammaDist make_prior_component_intensity](#) (double mean=[DefaultPriorMeanI](#), double kappa=[DefaultPriorIntensityKappa](#))
 - static [prior_hessian::TruncatedParetoDist make_prior_component_sigma](#) (double min_sigma, double max_↔ sigma, double alpha=[DefaultPriorPSFSigmaAlpha](#))
 - 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 `DefaultEstimatorMethod` = "TrustRegion"
 - Default optimization method for MLE/MAP estimation.*
- static const std::string `DefaultProfileBoundsEstimatorMethod` = "Newton"
 - Default optimization method for profile bounds optimizations.*
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const `IdxT` `DefaultMCMCNumSamples` = 300
 - Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)*
- static const `IdxT` `DefaultMCMCBurnin` = 10
 - Number of samples to throw away (burn-in) on initialization.*
- static const `IdxT` `DefaultMCMCThin` = 0
 - Keep every # samples. [Value of 0 indicates use the model default. This is suggested].*
- static const double `DefaultConfidenceLevel` = 0.95
 - Default level at which to estimate confidence intervals must be in range (0,1).*
- static const double `DefaultPriorBetaPos` = 3
- static const double `DefaultPriorSigmaPos` = 1
- static const double `DefaultPriorMeanI` = 300
- static const double `DefaultPriorMaxI` = INFINITY
- static const double `DefaultPriorIntensityKappa` = 2
- static const double `DefaultPriorPixelMeanBG` = 4
- static const double `DefaultPriorPSFSigmaAlpha` = 2
- 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 `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 & o)`

Definition at line 43 of file Gauss2DMap.cpp.

9.12.3.8 `mappel::Gauss2DMap::Gauss2DMap (Gauss2DMap && 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 & theta, double epsilon = bounds_epsilon) const`
[inherited]

Definition at line 255 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 & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 272 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 & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 313 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 & *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 & *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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.12.4.14 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`, `[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.12.4.25 `CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]`

Definition at line 207 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 211 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 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.12.4.30 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG`, `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::DefaultPriorPixelMeanBG, mappel::PointEmitterModel::make_prior_↵
 component_intensity(), mappel::PointEmitterModel::make_prior_component_position_normal(), and mappel::Image↵
 Format2DBase::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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanI, double kappa = DefaultPriorIntensityKappa) [static], [inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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().

9.12.4.56 **prior_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = DefaultPriorBetaPos) [static], [inherited]

Definition at line 99 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 = DefaultPriorSigmaPos) [static], [inherited]

Definition at line 92 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().

9.12.4.58 **prior_hessian::TruncatedParetoDist** **mappel::PointEmitterModel::make_prior_component_sigma** (*double min_sigma*, *double max_sigma*, *double alpha = DefaultPriorPSFSigmaAlpha*) [static],[inherited]

Definition at line 111 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 Gauss2DMap & mappel::Gauss2DMap::operator= (const Gauss2DMap & o)

Definition at line 59 of file Gauss2DMap.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::ImageFormat2DBase::operator=(), mappel::Gauss2DModel::operator=(), and mappel::PointEmitterModel::operator=().

9.12.4.62 Gauss2DMap & mappel::Gauss2DMap::operator= (Gauss2DMap && o)

Definition at line 70 of file Gauss2DMap.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::I(), 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::I(), 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 283 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 323 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 251 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 271 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.12.4.74 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline]`, `[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG, 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().

9.12.4.76 `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 220 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::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 267 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 247 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::DefaultPriorMeanI`, `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 233 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 259 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 117 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 244 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().

9.12.4.93 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const` `[inherited]`

Definition at line 264 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()`.

9.12.4.94 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const` `[inherited]`

Definition at line 303 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 67 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::DefaultConfidenceLevel = 0.95` `[static]`, `[inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.12.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion" [static],[inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.12.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10 [static],[inherited]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.12.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300 [static],[inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.12.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0 [static],[inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.12.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3 [static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.12.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2 [static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.12.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY [static],[inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.12.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300 [static],[inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.12.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.12.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.12.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.12.5.14 `const std::string mappel::Gauss2DModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 51 of file Gauss2DModel.h.

Referenced by `mappel::Gauss2DModel::get_psf_sigma()`.

9.12.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.12.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.12.5.17 `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` [static], [inherited]

Definition at line 23 of file PoissonNoise2DObjective.h.

9.12.5.18 `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()`.

9.12.5.19 `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()`.

9.12.5.20 `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()`.

9.12.5.21 `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()`.

9.12.5.22 `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()`.

9.12.5.23 `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()`.

9.12.5.24 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.12.5.25 `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()`.

9.12.5.26 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.12.5.27 `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()`.

9.12.5.28 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 155 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()`.

9.12.5.29 `const std::string mappel::Gauss2DMap::name` `[static]`

Definition at line 37 of file Gauss2DMap.h.

9.12.5.30 `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()`.

9.12.5.31 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.12.5.32 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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()`.

9.12.5.33 `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()`.

9.12.5.34 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 152 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().

9.12.5.35 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().

9.12.5.36 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().

9.12.5.37 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().

9.12.5.38 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().

9.12.5.39 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 155 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().

9.12.5.40 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().

9.12.5.41 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:

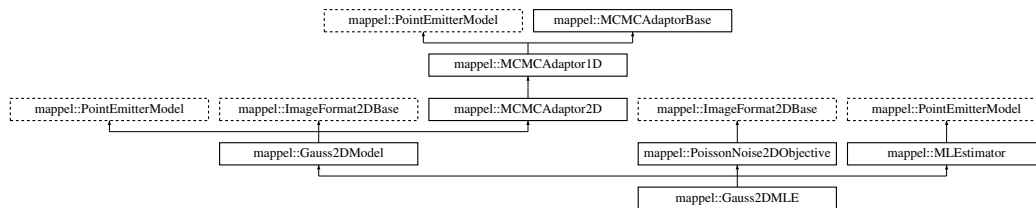
- [Gauss2DMAP.h](#)
- [Gauss2DMAP.cpp](#)

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_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_xpos, double pos_ypos, double sigma=DefaultPriorSigmaPos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_xpos, double pos_ypos, double beta=DefaultPriorBetaPos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=DefaultPriorMeanI, double kappa=DefaultPriorIntensityKappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_sigma, double alpha=DefaultPriorPSFSigmaAlpha)
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const IdxT [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const IdxT [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const IdxT [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 && *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 & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 255 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 & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 272 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 & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 313 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 & 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 & 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.13.4.14 **StringVecT** mappel::PointEmitterModel::get_hyperparam_names () const [inline], [inherited]

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.13.4.25 **CompositeDist &** mappel::PointEmitterModel::get_prior () [inline],[inherited]

Definition at line 207 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 211 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 127 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.13.4.30 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()` `[static],[inherited]`

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanI, double kappa = DefaultPriorIntensityKappa) [static], [inherited]`

Definition at line 105 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::DefaultPriorMaxI`.

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 = DefaultPriorBetaPos) [static], [inherited]`

Definition at line 99 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 = DefaultPriorSigmaPos) [static], [inherited]`

Definition at line 92 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()`.

9.13.4.58 **prior_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make_prior_component_sigma (double *min_sigma*, double *max_sigma*, double *alpha* = **DefaultPriorPSFSigmaAlpha**) [static],[inherited]

Definition at line 111 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>	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().

9.13.4.61 **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::I(), 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::I(), mappel::PointEmitterModel::↵
make_param(), mappel::Gauss2DModel::pixel_grad(), mappel::Gauss2DModel::psf_sigma, mappel::Gauss2D↵
Model::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 283 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 323 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 251 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 271 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.13.4.74 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline],[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG, 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().

9.13.4.76 `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 220 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::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 267 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 247 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_l = -1)` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::DefaultPriorMeanI`, `mappel::MCMCAdaptor1D::eta_l`, `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 233 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 259 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 117 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 244 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()`.

9.13.4.93 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const` `[inherited]`

Definition at line 264 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 303 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 67 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::DefaultConfidenceLevel = 0.95` `[static]`, `[inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.13.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion" [static],[inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.13.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10 [static],[inherited]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.13.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300 [static],[inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.13.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0 [static],[inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.13.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3 [static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.13.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2 [static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.13.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY [static],[inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.13.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300 [static],[inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.13.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.13.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.13.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.13.5.14 `const std::string mappel::Gauss2DModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 51 of file Gauss2DModel.h.

Referenced by `mappel::Gauss2DModel::get_psf_sigma()`.

9.13.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.13.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.13.5.17 `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` [static], [inherited]

Definition at line 23 of file PoissonNoise2DObjective.h.

9.13.5.18 `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()`.

9.13.5.19 `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()`.

9.13.5.20 `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()`.

9.13.5.21 `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()`.

9.13.5.22 `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()`.

9.13.5.23 `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()`.

9.13.5.24 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.13.5.25 `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()`.

9.13.5.26 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.13.5.27 `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()`.

9.13.5.28 `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 155 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()`.

9.13.5.29 `const std::string mappel::Gauss2DMLE::name` `[static]`

Definition at line 37 of file Gauss2DMLE.h.

9.13.5.30 `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()`.

9.13.5.31 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.13.5.32 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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()`.

9.13.5.33 `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()`.

9.13.5.34 CompositeDist mappel::PointEmitterModel::prior [protected], [inherited]

Definition at line 152 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().

9.13.5.35 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().

9.13.5.36 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().

9.13.5.37 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().

9.13.5.38 **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()`.

9.13.5.39 **ParamT** `mappel::PointEmitterModel::ubound` `[protected],[inherited]`

Definition at line 155 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()`.

9.13.5.40 **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()`.

9.13.5.41 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:

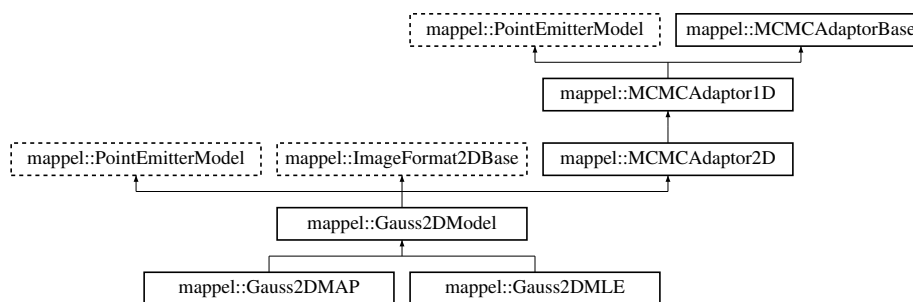
- [Gauss2DMLE.h](#)
- [Gauss2DMLE.cpp](#)

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` = `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 >`

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=DefaultPriorSigmaPos)
 - static prior_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta` (`IdxT` size, double pos_↵ beta=DefaultPriorBetaPos)
 - static prior_hessian::TruncatedGammaDist `make_prior_component_intensity` (double mean=DefaultPriorMeanI, double kappa=DefaultPriorIntensityKappa)
 - static prior_hessian::TruncatedParetoDist `make_prior_component_sigma` (double min_sigma, double max_↵ sigma, double alpha=DefaultPriorPSFSigmaAlpha)
 - 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 `DefaultEstimatorMethod` = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string `DefaultProfileBoundsEstimatorMethod` = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const `IdxT` `DefaultMCMCNumSamples` = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const `IdxT` `DefaultMCMCBurnin` = 10

Number of samples to throw away (burn-in) on initialization.

- static const `IdxT DefaultMCMCThin` = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested].
- static const double `DefaultConfidenceLevel` = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double `DefaultPriorBetaPos` = 3
- static const double `DefaultPriorSigmaPos` = 1
- static const double `DefaultPriorMeanI` = 300
- static const double `DefaultPriorMaxI` = INFINITY
- static const double `DefaultPriorIntensityKappa` = 2
- static const double `DefaultPriorPixelMeanBG` = 4
- static const double `DefaultPriorPSFSigmaAlpha` = 2
- 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 `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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.14.4.14 **StringVecT mappel::PointEmitterModel::get_hyperparam_names () const** [inline],[inherited]

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.14.4.25 CompositeDist & mappel::PointEmitterModel::get_prior () [inline], [inherited]

Definition at line 207 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 211 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 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.14.4.30 **ParallelRngManagerT** & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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* = **DefaultPriorMeanI**, double *kappa* = **DefaultPriorIntensityKappa**) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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* = **DefaultPriorBetaPos**) [static],[inherited]

Definition at line 99 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* = **DefaultPriorSigmaPos**) [static],[inherited]

Definition at line 92 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* = **DefaultPriorPSFSigmaAlpha**) [static],[inherited]

Definition at line 111 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 283 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 323 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 251 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 271 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.14.4.74 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline],[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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()`.

9.14.4.76 `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 220 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 267 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 247 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::DefaultPriorMeanI`, `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 233 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 259 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 117 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 244 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 264 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 303 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::ImageFormat2DBase::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 67 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::DefaultConfidenceLevel = 0.95` [static],[inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.14.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` [static],[inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.14.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static],[inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.14.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.14.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.14.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.14.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.14.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.14.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.14.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.14.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.14.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.14.5.14 `const std::string mappel::Gauss2DModel::DefaultPriorType = "Normal" [static]`

Definition at line 51 of file Gauss2DModel.h.

Referenced by `get_psf_sigma()`.

9.14.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton" [static],
[inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.14.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],
[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsxyModel::initial_theta_estimate()`.

9.14.5.17 `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()`.

9.14.5.18 `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()`.

9.14.5.19 `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()`.

9.14.5.20 `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()`.

9.14.5.21 `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()`.

9.14.5.22 `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()`.

9.14.5.23 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.14.5.24 `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()`.

9.14.5.25 `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 68 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.14.5.26 `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()`.

9.14.5.27 `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 155 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::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()`.

9.14.5.28 `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()`.

9.14.5.29 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.14.5.30 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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::Gauss1DModel::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()`.

9.14.5.31 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().

9.14.5.32 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 152 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().

9.14.5.33 const StringVecT mappel::Gauss2DModel::prior_types [static]

Initial value:

```
= { "Beta",
    "Normal"
}
```

Definition at line 50 of file Gauss2DModel.h.

Referenced by get_psf_sigma().

9.14.5.34 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().

9.14.5.35 `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()`.

9.14.5.36 `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::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()`, `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::ImageFormat2DBase::make_image()`, `mappel::ImageFormat2DBase::make_image_stack()`, `make_internal_1Dsum_estimator()`, `mappel::Gauss2DsModel::make_internal_1Dsum_estimator()`, `make_prior_beta_position()`, `mappel::Gauss2DsModel::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()`.

9.14.5.37 `ParamT mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 155 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()`, `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()`.

9.14.5.38 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().

9.14.5.39 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:

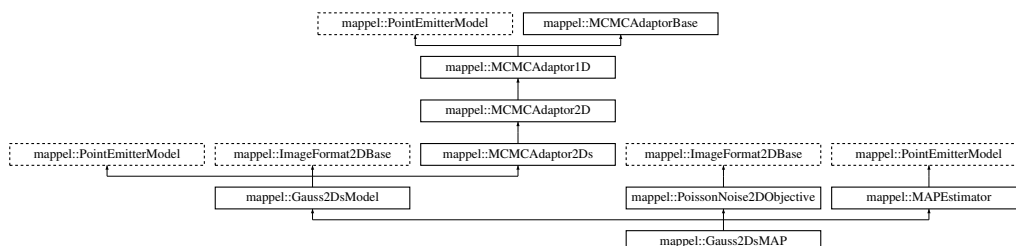
- [Gauss2DModel.h](#)
- [Gauss2DModel.cpp](#)

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=`DefaultPriorSigmaPos`)
- static prior_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta` (`IdxT` size, double pos_↵
beta=`DefaultPriorBetaPos`)
- static prior_hessian::TruncatedGammaDist `make_prior_component_intensity` (double mean=`DefaultPriorMeanI`,
double kappa=`DefaultPriorIntensityKappa`)
- static prior_hessian::TruncatedParetoDist `make_prior_component_sigma` (double min_sigma, double max_↵
sigma, double alpha=`DefaultPriorPSFSigmaAlpha`)
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.15.4.15 StringVecT mappel::PointEmitterModel::get_hyperparam_names () const [inline],[inherited]

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.15.4.31 `CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]`

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.15.4.33 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]`

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.15.4.34 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]`

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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* = **DefaultPriorMeanI**, double *kappa* = **DefaultPriorIntensityKappa**) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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* = **DefaultPriorBetaPos**) [static],[inherited]

Definition at line 99 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* = **DefaultPriorSigmaPos**) [static],[inherited]

Definition at line 92 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* = **DefaultPriorPSFSigmaAlpha**) [static],[inherited]

Definition at line 111 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::ImageFormat2DBase::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::Gauss2DsModel::Stencil::DY, mappel::Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::l(), mappel::Gauss2DsModel::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::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.

9.15.4.69 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::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.

9.15.4.70 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::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(), mappel::Gauss2DsModel::pixel_grad(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), mappel::Gauss2DsModel::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 283 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 323 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 251 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 271 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.15.4.78 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline]`, `[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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::DefaultPriorMeanI`, `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 233 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 259 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 117 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 244 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 264 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 303 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 67 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()`.

9.15.5.2 `const double mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95` `[static], [inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

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

9.15.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` `[static], [inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file `PointEmitterModel.h`.

9.15.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` `[static], [inherited]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file `PointEmitterModel.h`.

9.15.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` `[static], [inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file `PointEmitterModel.h`.

9.15.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` `[static], [inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file `PointEmitterModel.h`.

9.15.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file `PointEmitterModel.h`.

9.15.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.15.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.15.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.15.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.15.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.15.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.15.5.14 `const std::string mappel::Gauss2DsModel::DefaultPriorType = "Normal" [static],[inherited]`

Definition at line 59 of file Gauss2DsModel.h.

Referenced by `mappel::Gauss2DsModel::get_min_sigma()`.

9.15.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton" [static],[inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.15.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.15.5.17 `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names [static],[inherited]`

Definition at line 23 of file PoissonNoise2DObjective.h.

9.15.5.18 `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()`.

9.15.5.19 `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()`.

9.15.5.20 `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()`.

9.15.5.21 `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()`.

9.15.5.22 `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()`.

9.15.5.23 `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()`.

9.15.5.24 `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()`.

9.15.5.25 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.15.5.26 `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()`.

9.15.5.27 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.15.5.28 `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()`.

9.15.5.29 `ParamT mappel::PointEmitterModel::lbound` `[protected]`,`[inherited]`

Definition at line 155 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()`.

9.15.5.30 `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::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()`.

9.15.5.31 `const std::string mappel::Gauss2DsMAP::name` `[static]`

Definition at line 36 of file Gauss2DsMAP.h.

9.15.5.32 `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()`.

9.15.5.33 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.15.5.34 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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()`.

9.15.5.35 `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()`.

9.15.5.36 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 152 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().

9.15.5.37 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().

9.15.5.38 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().

9.15.5.39 **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()`.

9.15.5.40 **ParamT** `mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 155 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()`.

9.15.5.41 **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()`.

9.15.5.42 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:

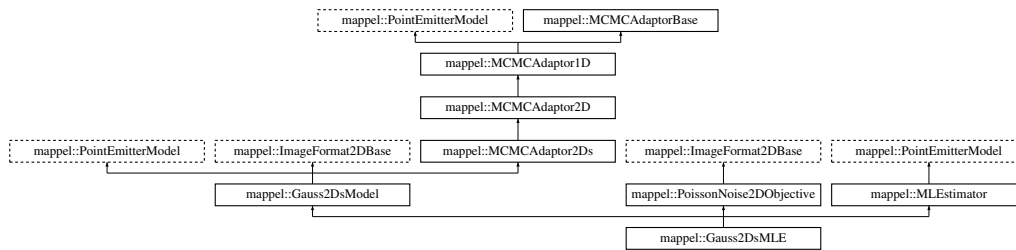
- [Gauss2DsMAP.h](#)
- [Gauss2DsMAP.cpp](#)

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=[DefaultPriorSigmaPos](#))
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↵beta=[DefaultPriorBetaPos](#))
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[DefaultPriorMeanI](#), double kappa=[DefaultPriorIntensityKappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵sigma, double alpha=[DefaultPriorPSFSigmaAlpha](#))
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double [DefaultConfidenceLevel](#) = 0.95

Default level at which to estimate confidence intervals must be in range (0,1).

- static const double `DefaultPriorBetaPos` = 3
- static const double `DefaultPriorSigmaPos` = 1
- static const double `DefaultPriorMeanI` = 300
- static const double `DefaultPriorMaxI` = INFINITY
- static const double `DefaultPriorIntensityKappa` = 2
- static const double `DefaultPriorPixelMeanBG` = 4
- static const double `DefaultPriorPSFSigmaAlpha` = 2
- 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 `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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.16.4.15 **StringVecT** mappel::PointEmitterModel::get_hyperparam_names () const [inline], [inherited]

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.16.4.31 CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.16.4.33 ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.16.4.34 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 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 223 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxxyModel::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 235 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::DefaultPriorPixelMeanBG`, `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::DefaultPriorPixelMeanBG`, `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::Gauss2DsModel()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

9.16.4.50 `PointEmitterModel::ParamT` `mappel::PointEmitterModel::make_param () const` `[inline], [inherited]`

Definition at line 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanI, double kappa = DefaultPriorIntensityKappa)` `[static]`, `[inherited]`

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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 = DefaultPriorBetaPos*) [static],[inherited]

Definition at line 99 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 = DefaultPriorSigmaPos*) [static],[inherited]

Definition at line 92 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 = DefaultPriorPSFSigmaAlpha*) [static],[inherited]

Definition at line 111 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::ImageFormat2DBase::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::ImageFormat2DBase::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::Gauss2DsModel::Stencil::DY`, `mappel::Gauss2DsModel::Stencil::DYS`, `mappel::Gauss2DsModel::Stencil::l()`, `mappel::Gauss2DsModel::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::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`.

9.16.4.69 `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::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`.

9.16.4.70 `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::l()`, `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::l()`, `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 283 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 323 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 251 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 271 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.16.4.78 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const [inline],[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG, 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().

9.16.4.80 `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 220 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::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 267 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 247 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::DefaultPriorMeanI`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::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 233 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()`.

9.16.4.87 `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 259 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 117 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 244 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 264 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::Gauss2DsxyModel::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 303 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 67 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::DefaultConfidenceLevel = 0.95 [static], [inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.16.5.3 const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion" [static], [inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.16.5.4 const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10 [static], [inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.16.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` `[static], [inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.16.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` `[static], [inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested].

Definition at line 56 of file PointEmitterModel.h.

9.16.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.16.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.16.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.16.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.16.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.16.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.16.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.16.5.14 `const std::string mappel::Gauss2DsModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 59 of file Gauss2DsModel.h.

Referenced by `mappel::Gauss2DsModel::get_min_sigma()`.

9.16.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.16.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.16.5.17 `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` `[static], [inherited]`

Definition at line 23 of file PoissonNoise2DObjective.h.

9.16.5.18 `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()`.

9.16.5.19 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().

9.16.5.20 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().

9.16.5.21 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().

9.16.5.22 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().

9.16.5.23 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().

9.16.5.24 `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()`.

9.16.5.25 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.16.5.26 `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()`.

9.16.5.27 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.16.5.28 `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()`.

9.16.5.29 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 155 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()`.

9.16.5.30 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::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().

9.16.5.31 const std::string mappel::Gauss2DsMLE::name [static]

Definition at line 36 of file Gauss2DsMLE.h.

9.16.5.32 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().

9.16.5.33 IdxT mappel::PointEmitterModel::num_hyperparams [protected],[inherited]

Definition at line 154 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get_num_hyperparams(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set_prior().

9.16.5.34 IdxT mappel::PointEmitterModel::num_params [protected],[inherited]

Definition at line 153 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().

9.16.5.35 `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()`.

9.16.5.36 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 152 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()`.

9.16.5.37 `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()`.

9.16.5.38 `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()`.

9.16.5.39 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().

9.16.5.40 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 155 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().

9.16.5.41 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().

9.16.5.42 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:

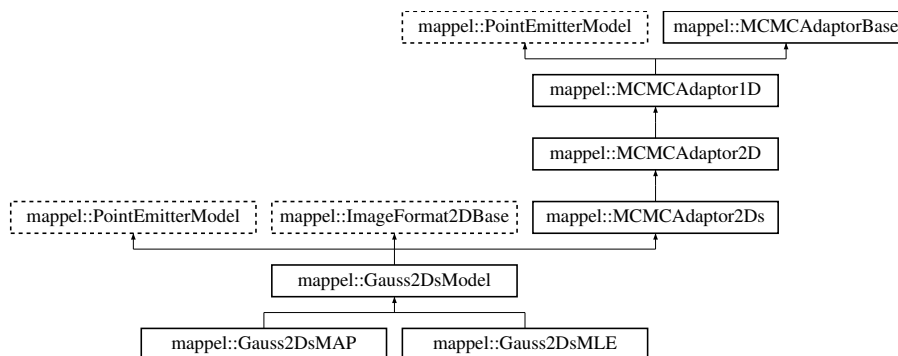
- [Gauss2DsMLE.h](#)
- [Gauss2DsMLE.cpp](#)

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=DefaultPriorSigmaPos)
- static prior_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta` (IdxT size, double pos_↵
beta=DefaultPriorBetaPos)
- static prior_hessian::TruncatedGammaDist `make_prior_component_intensity` (double mean=DefaultPriorMeanI, double kappa=DefaultPriorIntensityKappa)
- static prior_hessian::TruncatedParetoDist `make_prior_component_sigma` (double min_sigma, double max_↵
sigma, double alpha=DefaultPriorPSFSigmaAlpha)
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.17.4.15 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`, `[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.17.4.31 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline],[inherited]`

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.17.4.33 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static],[inherited]`

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.17.4.34 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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(), 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanI, double kappa = DefaultPriorIntensityKappa) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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 = DefaultPriorBetaPos) [static],[inherited]

Definition at line 99 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 = DefaultPriorSigmaPos) [static],[inherited]

Definition at line 92 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 = DefaultPriorPSFSigmaAlpha) [static],[inherited]

Definition at line 111 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::I()`, `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 283 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 323 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 251 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 271 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.17.4.78 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const [inline], [inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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::DefaultPriorMeanI`, `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 233 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 259 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 117 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 244 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 264 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 303 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 67 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::DefaultConfidenceLevel = 0.95` `[static], [inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.17.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` `[static], [inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.17.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static],[inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.17.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.17.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.17.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.17.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.17.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.17.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` [static],[inherited]

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.17.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.17.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static],[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.17.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.17.5.14 `const std::string mappel::Gauss2DsModel::DefaultPriorType = "Normal"` `[static]`

Definition at line 59 of file Gauss2DsModel.h.

Referenced by `get_min_sigma()`.

9.17.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static],[inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.17.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static],[inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `initial_theta_estimate()`.

9.17.5.17 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().

9.17.5.18 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().

9.17.5.19 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().

9.17.5.20 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().

9.17.5.21 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().

9.17.5.22 `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()`.

9.17.5.23 `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()`.

9.17.5.24 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.17.5.25 `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()`.

9.17.5.26 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.17.5.27 `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()`.

9.17.5.28 ParamT mappel::PointEmitterModel::lbound [protected],[inherited]

Definition at line 155 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().

9.17.5.29 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().

9.17.5.30 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().

9.17.5.31 IdxT mappel::PointEmitterModel::num_hyperparams [protected],[inherited]

Definition at line 154 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get_num_hyperparams(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set_prior().

9.17.5.32 IdxT mappel::PointEmitterModel::num_params [protected],[inherited]

Definition at line 153 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(), 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().

9.17.5.33 `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()`.

9.17.5.34 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 152 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()`.

9.17.5.35 `const StringVecT mappel::Gauss2DsModel::prior_types` `[static]`

Initial value:

```
= { "Beta",
    "Normal"
}
```

Definition at line 58 of file `Gauss2DsModel.h`.

Referenced by `get_min_sigma()`.

9.17.5.36 `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()`.

9.17.5.37 `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::Gauss2D←Model::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::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()`, `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::Gauss2←DModel::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()`.

9.17.5.38 `ParamT mappel::PointEmitterModel::ubound` [protected], [inherited]

Definition at line 155 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::Point←EmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

9.17.5.39 `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()`, `set_size()`, and `update_internal_1Dsum_estimators()`.

9.17.5.40 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(), set_size(), and update_internal_1Dsum_estimators().

The documentation for this class was generated from the following files:

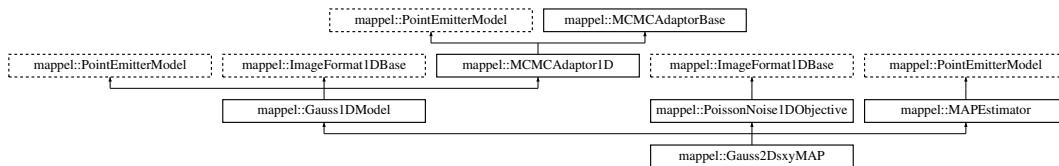
- [Gauss2DsModel.h](#)
- [Gauss2DsModel.cpp](#)

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=DefaultPriorSigmaPos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_↵ beta=DefaultPriorBetaPos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=DefaultPriorMeanI, double kappa=DefaultPriorIntensityKappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵ sigma, double alpha=DefaultPriorPSFSigmaAlpha)
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const IdxT [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const IdxT [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const IdxT [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.18.4.12 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`, `[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.18.4.23 CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]

Definition at line 207 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 211 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 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.18.4.28 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 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 223 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 235 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::DefaultPriorPixelMeanBG, 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::DefaultPriorPixelMeanBG, 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanl, double kappa = DefaultPriorIntensityKappa)` `[static]`, `[inherited]`

Definition at line 105 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::DefaultPriorMaxl`.

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 = DefaultPriorBetaPos)` `[static]`, `[inherited]`

Definition at line 99 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()`.

9.18.4.53 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (*IdxT size*, *double pos_sigma = DefaultPriorSigmaPos*) [static],[inherited]

Definition at line 92 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 = DefaultPriorPSFSigmaAlpha*) [static],[inherited]

Definition at line 111 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 283 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 323 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 251 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 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.18.4.68 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline],[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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 227 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::DefaultPriorMeanI`, `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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.18.4.80 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` [inherited]

Definition at line 165 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 156 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 117 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 244 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 264 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 303 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 67 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::DefaultConfidenceLevel = 0.95` [static], [inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.18.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` [static], [inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.18.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static], [inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.18.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.18.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested].

Definition at line 56 of file PointEmitterModel.h.

9.18.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.18.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.18.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.18.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` [static],[inherited]

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.18.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` [static],[inherited]

Default per-pixel mean background counts

Definition at line 64 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()`.

9.18.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.18.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.18.5.14 `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal"` [static], [inherited]

Definition at line 53 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::operator=()`.

9.18.5.15 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` [static], [inherited]

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.18.5.16 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` [static], [inherited]

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.18.5.17 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names` [static], [inherited]

Definition at line 24 of file PoissonNoise1DObjective.h.

9.18.5.18 `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()`.

9.18.5.19 `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()`.

9.18.5.20 `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()`.

9.18.5.21 `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()`.

9.18.5.22 `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()`.

9.18.5.23 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.18.5.24 `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()`.

9.18.5.25 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.18.5.26 `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()`.

9.18.5.27 `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 155 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()`.

9.18.5.28 `const std::string mappel::Gauss2DsxyMAP::name` `[static]`

Definition at line 30 of file Gauss2DsxyMAP.h.

9.18.5.29 `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()`.

9.18.5.30 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.18.5.31 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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()`.

9.18.5.32 `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()`.

9.18.5.33 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 152 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()`.

9.18.5.34 `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=()`.

9.18.5.35 `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()`.

9.18.5.36 `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()`.

9.18.5.37 `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()`.

9.18.5.38 `ParamT mappel::PointEmitterModel::ubound` `[protected]`, `[inherited]`

Definition at line 155 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:

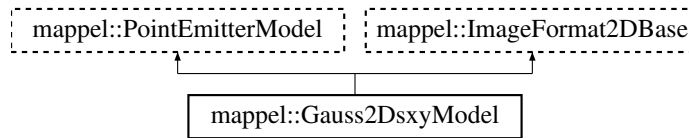
- [Gauss2DsxyMAP.h](#)

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=DefaultPriorSigmaPos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↵beta=DefaultPriorBetaPos)

- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[DefaultPriorMeanI](#), double kappa=[DefaultPriorIntensityKappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double [min_sigma](#), double max_↵sigma, double alpha=[DefaultPriorPSFSigmaAlpha](#))
- 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 [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested].
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 & 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 255 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 272 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 313 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 174 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 183 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 192 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 204 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::Gauss2DsxModel::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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.19.4.13 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const [inline],[inherited]`

Definition at line 263 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 239 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 231 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 219 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::Gauss2DsxxyModel::get_max_sigma () const** [inline]

Definition at line 127 of file Gauss2DsxxyModel.h.

References get_max_sigma_ratio(), and get_min_sigma().

9.19.4.19 **double mappel::Gauss2DsxxyModel::get_max_sigma (IdxT dim) const** [inline]

Definition at line 131 of file Gauss2DsxxyModel.h.

References get_max_sigma_ratio(), and get_min_sigma().

9.19.4.20 **double mappel::Gauss2DsxxyModel::get_max_sigma_ratio () const** [inline]

Definition at line 135 of file Gauss2DsxxyModel.h.

References mappel::PointEmitterModel::get_ubound().

Referenced by get_max_sigma().

9.19.4.21 **VecT mappel::Gauss2DsxxyModel::get_min_sigma () const** [inline]

Definition at line 122 of file Gauss2DsxxyModel.h.

References min_sigma.

Referenced by get_max_sigma().

9.19.4.22 **double mappel::Gauss2DsxxyModel::get_min_sigma (IdxT dim) const**

9.19.4.23 **IdxT mappel::PointEmitterModel::get_num_hyperparams () const** [inline], [inherited]

Definition at line 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.27 CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.19.4.29 ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.19.4.30 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 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 223 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 235 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 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 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 = DefaultPriorMeanl*, *double kappa = DefaultPriorIntensityKappa*) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxl.

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 = DefaultPriorBetaPos) [static],[inherited]`

Definition at line 99 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()`.

9.19.4.54 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal (IdxT size, double pos_sigma = DefaultPriorSigmaPos) [static],[inherited]`

Definition at line 92 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()`.

9.19.4.55 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma (double min_sigma, double max_sigma, double alpha = DefaultPriorPSFSigmaAlpha) [static],[inherited]`

Definition at line 111 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>	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 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 283 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 323 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 251 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 271 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.19.4.68 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline], [inherited]`

Definition at line 275 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 220 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 267 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 247 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 233 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 259 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 156 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 117 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 244 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 264 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::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 303 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 67 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::DefaultConfidenceLevel = 0.95` [static],[inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.19.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` [static],[inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.19.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static],[inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.19.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.19.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.19.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.19.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.19.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.19.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.19.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.19.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.19.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.19.5.14 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.19.5.15 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.19.5.16 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.19.5.17 `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()`.

9.19.5.18 `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 68 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.19.5.19 `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()`.

9.19.5.20 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 155 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()`.

9.19.5.21 `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`.

9.19.5.22 `double mappel::Gauss2DsxModel::mcmc_candidate_eta_y` [protected]

Std-dev for the normal perturbations to `theta_y` under MCMC sampling

Definition at line 107 of file `Gauss2DsxModel.h`.

9.19.5.23 `VecT mappel::Gauss2DsxModel::min_sigma` [protected]

Gaussian PSF in pixels

Definition at line 113 of file `Gauss2DsxModel.h`.

Referenced by `get_min_sigma()`.

9.19.5.24 `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()`.

9.19.5.25 `IdxT mappel::PointEmitterModel::num_hyperparams` [protected],[inherited]

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

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.19.5.26 `IdxT mappel::PointEmitterModel::num_params` [protected],[inherited]

Definition at line 153 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()`.

9.19.5.27 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 152 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().

9.19.5.28 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().

9.19.5.29 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 155 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().

9.19.5.30 Gauss1DsMAP mappel::Gauss2DskyModel::x_model [protected]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 114 of file Gauss2DskyModel.h.

9.19.5.31 Gauss1DsMAP mappel::Gauss2DskyModel::y_model [protected]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 115 of file Gauss2DskyModel.h.

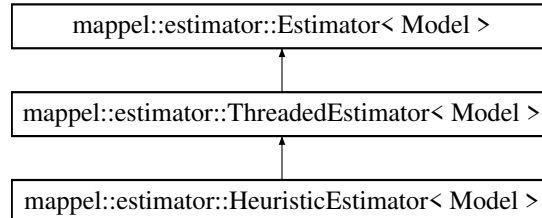
The documentation for this class was generated from the following file:

- [Gauss2DskyModel.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, [_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, [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 334 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 337 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_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\(\)](#).

9.20.3.6 `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>	MLEData 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>	MLEData 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>	MLEData 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>	MLEDebugData 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.

Parameters

out	<i>mle_data</i>	MLEData 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>	MLEData 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](#), [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`s, `mappel::estimator::ProfileBoundsData::initialize_arrays()`, `mappel←
::estimator::ProfileBoundsData::mle`, `mappel::estimator::Estimator< Model >::model`, `mappel::estimator::Profile←
BoundsData::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::ML←
EData::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()`.

9.20.3.16 `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`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::ProfileBounds←
DataStack::Nparams_est`, `mappel::estimator::ThreadedEstimator< Model >::num_threads`, `mappel::estimator::ML←
EData::obsI`, `mappel::estimator::MLEDataStack::obsI`, `mappel::print_text_image()`, `mappel::estimator::ProfileBounds←
Data::profile_lb`, `mappel::estimator::ProfileBoundsDataStack::profile_lb`, `mappel::estimator::ProfileBoundsData←
::profile_points_lb`, `mappel::estimator::ProfileBoundsDataStack::profile_points_lb`, `mappel::estimator::ProfileBounds←
Data::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::MLEData←
Stack::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_idxes, 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_idxes`, `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 276 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 341 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 301 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 326 of file estimator.h.

9.20.4.3 `template<class Model> const Model& mappel::estimator::Estimator< Model >::model` [protected], [inherited]

Definition at line 296 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 328 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 299 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 327 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 300 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:

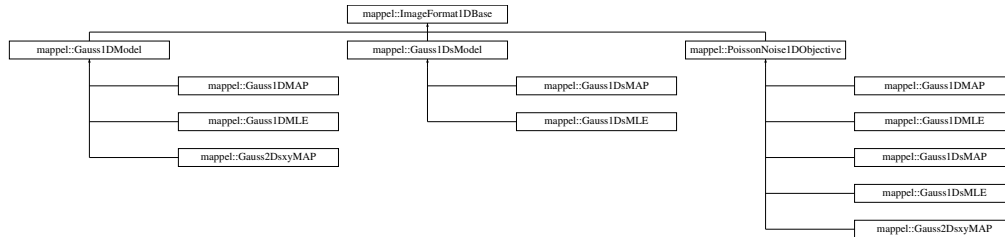
- [estimator.h](#)
- [estimator_impl.h](#)

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:

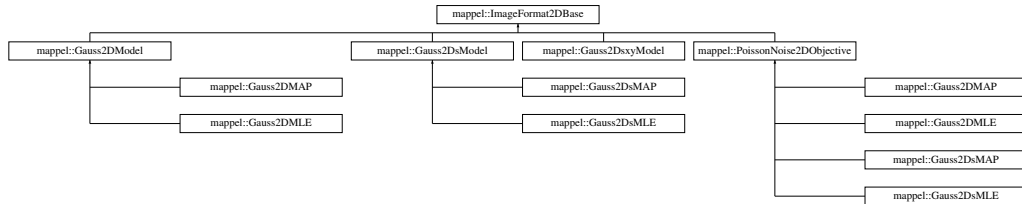
- [ImageFormat1DBase.h](#)
- [ImageFormat1DBase.cpp](#)

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::Gauss2DMLC←
::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()`.

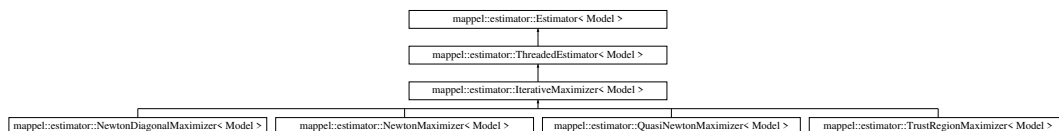
The documentation for this class was generated from the following files:

- [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, [MLEData](#) &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
- 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, [MLEData](#) &data, [StencilT](#)< [Model](#) > &mle_stencil)
- void [estimate_max](#) (const [ModelDataT](#)< [Model](#) > &data, const [ParamT](#)< [Model](#) > &theta_init, [MLEData](#) &mle, [MLEData](#) &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, [MLEData](#) &mle_data, [StencilT](#)< [Model](#) > &mle_stencil)
- void [estimate_max_debug](#) (const [ModelDataT](#)< [Model](#) > &data, const [ParamT](#)< [Model](#) > &theta_init, [MLEData](#) &mle_data, [MLEData](#) &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, [ProfileBoundsData](#) &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, [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_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 426 of file estimator.h.

9.23.2 Constructor & 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_idxs, 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>	MLEData 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 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>	MLEData 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 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>	MLEData 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 ModelIDataT< 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>	MLEDebugData 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>	MLEData 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>	MLEData 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, 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, 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_idxxs`, `mappel::estimator::ProfileBoundsDataStack::estimated_idxxs`, `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::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()`.

9.23.3.21 `template<class Model > double mappel::estimator::Estimator< Model >::estimate_profile_max (const ModelDataT< Model > & data, const IdxVecT & fixed_idxxs, 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_idxxs`, `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 276 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 443 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 445 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 444 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 442 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::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()`.

9.23.3.34 `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_idx()`, `mappel::estimator::IterativeMaximizer< Model >::MaximizerData::has_theta_sequence()`, `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idx`, `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 462 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 463 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 461 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 458 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 459 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 430 of file estimator.h.

9.23.4.7 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts [protected], [inherited]`

Definition at line 301 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 477 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 464 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 468 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 326 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 456 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 466 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 296 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 328 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 299 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 327 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 472 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 474 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 473 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 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()`.

9.23.4.22 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected]`,
`[inherited]`

Definition at line 300 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:

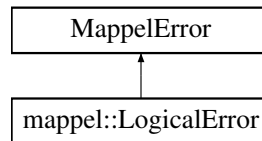
- [estimator.h](#)
- [estimator_impl.h](#)

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:

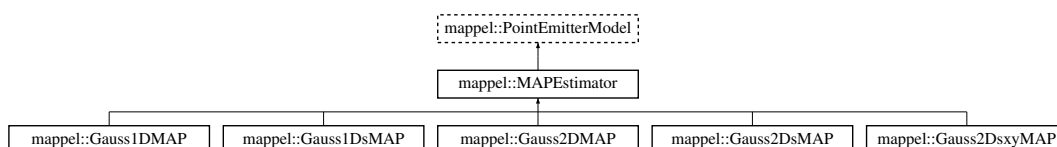
- [util.h](#)

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=`DefaultPriorSigmaPos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos_↵ beta=`DefaultPriorBetaPos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=`DefaultPriorMeanI`, double kappa=`DefaultPriorIntensityKappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min_sigma, double max_↵ sigma, double alpha=`DefaultPriorPSFSigmaAlpha`)
- static void `set_rng_seed` (`RngSeedT` seed)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

Static Public Attributes

- static const std::string `DefaultEstimatorMethod` = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string `DefaultProfileBoundsEstimatorMethod` = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const `IdxT` `DefaultMCMCNumSamples` = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const `IdxT` `DefaultMCMCBurnin` = 10
Number of samples to throw away (burn-in) on initialization.
- static const `IdxT` `DefaultMCMCThin` = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].
- static const double `DefaultConfidenceLevel` = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double `DefaultPriorBetaPos` = 3
- static const double `DefaultPriorSigmaPos` = 1
- static const double `DefaultPriorMeanI` = 300
- static const double `DefaultPriorMaxI` = INFINITY
- static const double `DefaultPriorIntensityKappa` = 2
- static const double `DefaultPriorPixelMeanBG` = 4
- static const double `DefaultPriorPSFSigmaAlpha` = 2
- 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

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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.25.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` [inline], [inherited]

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.25.4.16 `CompositeDist & mappel::PointEmitterModel::get_prior () [inline], [inherited]`

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.25.4.18 ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.25.4.19 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.25.4.20 StatsT mappel::PointEmitterModel::get_stats () const [inherited]

Definition at line 132 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 223 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 235 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.25.4.23 PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const [inline],[inherited]

Definition at line 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.25.4.31 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double *mean* = **DefaultPriorMeanI**, double *kappa* = **DefaultPriorIntensityKappa**) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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().

9.25.4.32 **prior_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make_prior_component_position_beta (*IdxT size*, double *pos_beta* = **DefaultPriorBetaPos**) [static],[inherited]

Definition at line 99 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().

9.25.4.33 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (*IdxT size*, double *pos_sigma* = **DefaultPriorSigmaPos**) [static],[inherited]

Definition at line 92 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().

9.25.4.34 **prior_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make_prior_component_sigma (double *min_sigma*, double *max_sigma*, double *alpha* = **DefaultPriorPSFSigmaAlpha**) [static],[inherited]

Definition at line 111 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::Gauss1DMap::operator=()`, `mappel::Gauss1DsMap::operator=()`, `mappel::Gauss2DsMap::operator=()`, and `mappel::Gauss2DMap::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 283 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 323 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 251 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 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.25.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline],[inherited]`

Definition at line 275 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 220 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 267 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 247 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 227 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 233 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 259 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.25.4.48 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_) [inherited]`

Definition at line 165 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 156 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 117 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.25.4.51 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 244 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 264 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 303 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 67 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()`.

9.25.5.2 `const double mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95` `[static], [inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

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

9.25.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` `[static], [inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file `PointEmitterModel.h`.

9.25.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` `[static], [inherited]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file `PointEmitterModel.h`.

9.25.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` `[static], [inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file `PointEmitterModel.h`.

9.25.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` `[static], [inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested].

Definition at line 56 of file `PointEmitterModel.h`.

9.25.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file `PointEmitterModel.h`.

9.25.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.25.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.25.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.25.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.25.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.25.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.25.5.14 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton" [static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.25.5.15 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.25.5.16 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.25.5.17 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.25.5.18 `ParamT mappel::PointEmitterModel::lbound [protected], [inherited]`

Definition at line 155 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()`.

9.25.5.19 `IdxT mappel::PointEmitterModel::num_hyperparams [protected], [inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.25.5.20 **IdxT** `mappel::PointEmitterModel::num_params` [protected],[inherited]

Definition at line 153 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()`.

9.25.5.21 **CompositeDist** `mappel::PointEmitterModel::prior` [protected],[inherited]

Definition at line 152 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()`.

9.25.5.22 **ParamT** `mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 155 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](#)s () 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 479 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](#), [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 506 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 504 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 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()`.

9.26.3.5 `template<class Model > VecT mappel::estimator::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh () const [inline]`

Definition at line 507 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 530 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 503 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 531 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 498 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`, `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 494 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 521 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 516 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 523 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 527 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 511 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 510 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 525 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_idxxs [protected]`

Definition at line 542 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 536 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 perpare to save if debugging.

Definition at line 533 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 489 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 489 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 482 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 481 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 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()`.

9.26.4.9 **template<class Model > int mappel::estimator::IterativeMaximizer< Model >::MaximizerData::nBacktracks =0**

Definition at line 486 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 487 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 534 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 484 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 535 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 535 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 539 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 541 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 483 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 540 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()`.

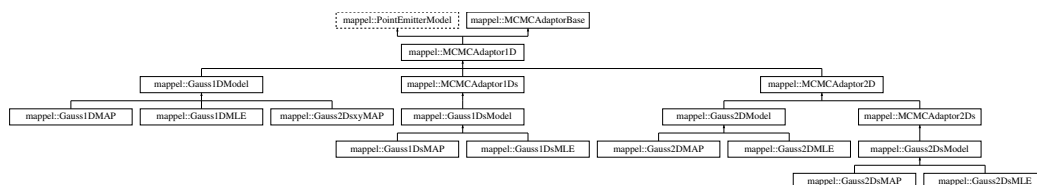
The documentation for this class was generated from the following files:

- [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=`DefaultPriorSigmaPos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` `size`, double `pos`↵
beta=`DefaultPriorBetaPos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double `mean`=`DefaultPriorMeanI`,
double `kappa`=`DefaultPriorIntensityKappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double `min_sigma`, double `max_`↵
sigma, double `alpha`=`DefaultPriorPSFSigmaAlpha`)
- static void `set_rng_seed` (`RngSeedT` `seed`)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

Static Public Attributes

- static const std::string `DefaultEstimatorMethod` = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string `DefaultProfileBoundsEstimatorMethod` = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const `IdxT` `DefaultMCMCNumSamples` = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const `IdxT` `DefaultMCMCBurnin` = 10
Number of samples to throw away (burn-in) on initialization.
- static const `IdxT` `DefaultMCMCThin` = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double `DefaultConfidenceLevel` = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double `DefaultPriorBetaPos` = 3
- static const double `DefaultPriorSigmaPos` = 1
- static const double `DefaultPriorMeanI` = 300
- static const double `DefaultPriorMaxI` = INFINITY
- static const double `DefaultPriorIntensityKappa` = 2

- static const double `DefaultPriorPixelMeanBG` = 4
- static const double `DefaultPriorPSFSigmaAlpha` = 2
- 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 `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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.27.4.9 StringVecT mappel::PointEmitterModel::get_hyperparam_names () const [inline],[inherited]

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.18 `CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]`

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.20 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]`

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.27.4.21 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]`

Definition at line 122 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 223 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 235 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const` `[inline]`,`[inherited]`

Definition at line 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.27.4.33 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double *mean* = DefaultPriorMeanI, double *kappa* = DefaultPriorIntensityKappa) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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().

9.27.4.34 **prior_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make_prior_component_position_beta (*IdxT size*, *double pos_beta = DefaultPriorBetaPos*) [static],[inherited]

Definition at line 99 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().

9.27.4.35 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (*IdxT size*, *double pos_sigma = DefaultPriorSigmaPos*) [static],[inherited]

Definition at line 92 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 = DefaultPriorPSFSigmaAlpha*) [static],[inherited]

Definition at line 111 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 283 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 323 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 251 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 271 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.27.4.45 PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const `[inline], [inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 & lboud_, const ParamT & ubound_)`
[*inherited*]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 220 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lboud`, `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 267 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 247 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 227 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::DefaultPriorMeanI`, `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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.27.4.56 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 165 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()`.

9.27.4.57 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 156 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::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 117 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.27.4.59 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 244 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().

9.27.4.60 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const [inherited]`

Definition at line 264 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 303 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 67 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::DefaultConfidenceLevel = 0.95 [static],[inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.27.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion" [static],[inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.27.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10 [static],[inherited]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.27.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300 [static],[inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.27.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0 [static],[inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.27.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3 [static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.27.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2 [static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.27.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY [static],[inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.27.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300 [static],[inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `set_intensity_mcmc_sampling()`.

9.27.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.27.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.27.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.27.5.14 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.27.5.15 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.27.5.16 `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()`.

9.27.5.17 `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()`.

9.27.5.18 `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()`.

9.27.5.19 `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()`.

9.27.5.20 `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()`.

9.27.5.21 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.27.5.22 `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 68 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.27.5.23 ParamT mappel::PointEmitterModel::lbound [protected], [inherited]

Definition at line 155 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().

9.27.5.24 IdxT mappel::PointEmitterModel::num_hyperparams [protected], [inherited]

Definition at line 154 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::get_num_hyperparams(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::operator=(), and mappel::PointEmitterModel::set_prior().

9.27.5.25 IdxT mappel::PointEmitterModel::num_params [protected], [inherited]

Definition at line 153 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().

9.27.5.26 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(), sample_mcmc_candidate(), and mappel::MCMCAdaptorBase::set_mcmc_num_phases().

9.27.5.27 CompositeDist mappel::PointEmitterModel::prior [protected], [inherited]

Definition at line 152 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().

9.27.5.28 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(), 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().

9.27.5.29 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 155 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().

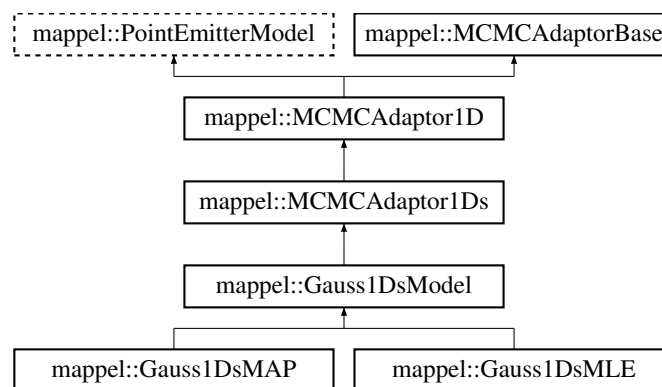
The documentation for this class was generated from the following files:

- [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_↵, sigma=`DefaultPriorSigmaPos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos_↵, beta=`DefaultPriorBetaPos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=`DefaultPriorMeanI`, double kappa=`DefaultPriorIntensityKappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min_sigma, double max_↵, sigma, double alpha=`DefaultPriorPSFSigmaAlpha`)
- static void `set_rng_seed` (`RngSeedT` seed)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

Static Public Attributes

- static const std::string `DefaultEstimatorMethod` = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string `DefaultProfileBoundsEstimatorMethod` = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const `IdxT` `DefaultMCMCNumSamples` = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const `IdxT` `DefaultMCMCBurnin` = 10
Number of samples to throw away (burn-in) on initialization.
- static const `IdxT` `DefaultMCMCThin` = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested].
- static const double `DefaultConfidenceLevel` = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double `DefaultPriorBetaPos` = 3
- static const double `DefaultPriorSigmaPos` = 1
- static const double `DefaultPriorMeanI` = 300
- static const double `DefaultPriorMaxI` = INFINITY
- static const double `DefaultPriorIntensityKappa` = 2
- static const double `DefaultPriorPixelMeanBG` = 4
- static const double `DefaultPriorPSFSigmaAlpha` = 2
- 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 `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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.28.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline],[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.28.4.18 `CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]`

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.28.4.20 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]`

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.28.4.21 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]`

Definition at line 122 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 223 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().

9.28.4.24 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline], [inherited]`

Definition at line 235 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.28.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const [inline], [inherited]`

Definition at line 171 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().

9.28.4.26 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const [inherited]`

Definition at line 188 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 179 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 198 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 183 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 203 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 175 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 193 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.28.4.33 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double *mean* = DefaultPriorMeanI, double *kappa* = DefaultPriorIntensityKappa) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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 = DefaultPriorBetaPos*) [static],[inherited]

Definition at line 99 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()**.

9.28.4.35 **prior_hessian::TruncatedNormalDist** **mappel::PointEmitterModel::make_prior_component_position_normal** (*IdxT size*, *double pos_sigma = DefaultPriorSigmaPos*) [static],[inherited]

Definition at line 92 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()**.

9.28.4.36 **prior_hessian::TruncatedParetoDist** **mappel::PointEmitterModel::make_prior_component_sigma** (*double min_sigma*, *double max_sigma*, *double alpha = DefaultPriorPSFSigmaAlpha*) [static],[inherited]

Definition at line 111 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 283 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 323 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 251 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 271 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.28.4.45 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline],[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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 227 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_l = -1)` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::DefaultPriorMeanI`, `mappel::MCMCAdaptor1D::eta_l`, `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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.28.4.56 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` [inherited]

Definition at line 165 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 156 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 117 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.28.4.59 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 244 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 264 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 303 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 67 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()`.

9.28.5.2 `const double mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95` [static],[inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.28.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` [static],[inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.28.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static],[inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.28.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.28.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.28.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.28.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.28.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.28.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.28.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.28.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.28.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.28.5.14 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.28.5.15 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.28.5.16 `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()`.

9.28.5.17 `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()`.

9.28.5.18 `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()`.

9.28.5.19 `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()`.

9.28.5.20 `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()`.

9.28.5.21 `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()`.

9.28.5.22 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.28.5.23 `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 68 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.28.5.24 `ParamT mappel::PointEmitterModel::lbound` `[protected],[inherited]`

Definition at line 155 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()`.

9.28.5.25 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected],[inherited]`

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

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.28.5.26 `IdxT mappel::PointEmitterModel::num_params` `[protected],[inherited]`

Definition at line 153 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()`.

9.28.5.27 `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()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

9.28.5.28 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 152 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()`.

9.28.5.29 `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()`, `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()`.

9.28.5.30 `ParamT mappel::PointEmitterModel::ubound` `[protected]`, `[inherited]`

Definition at line 155 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()`.

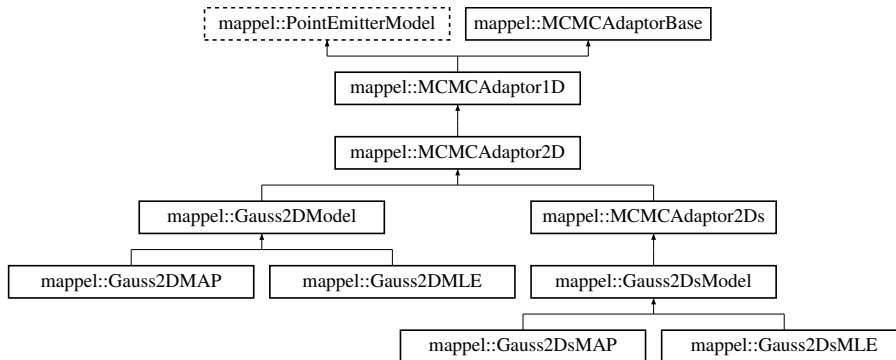
The documentation for this class was generated from the following files:

- [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=`DefaultPriorSigmaPos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos_↔ beta=`DefaultPriorBetaPos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=`DefaultPriorMeanI`, double kappa=`DefaultPriorIntensityKappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min_sigma, double max_↔ sigma, double alpha=`DefaultPriorPSFSigmaAlpha`)
- static void `set_rng_seed` (`RngSeedT` seed)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

Static Public Attributes

- static const std::string `DefaultEstimatorMethod` = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string `DefaultProfileBoundsEstimatorMethod` = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const `IdxT` `DefaultMCMCNumSamples` = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const `IdxT` `DefaultMCMCBurnin` = 10
Number of samples to throw away (burn-in) on initialization.
- static const `IdxT` `DefaultMCMCThin` = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double `DefaultConfidenceLevel` = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double `DefaultPriorBetaPos` = 3
- static const double `DefaultPriorSigmaPos` = 1
- static const double `DefaultPriorMeanI` = 300
- static const double `DefaultPriorMaxI` = INFINITY
- static const double `DefaultPriorIntensityKappa` = 2
- static const double `DefaultPriorPixelMeanBG` = 4
- static const double `DefaultPriorPSFSigmaAlpha` = 2
- 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 `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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.29.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` [inline], [inherited]

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.29.4.18 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline]`, `[inherited]`

Definition at line 207 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 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.29.4.20 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static],[inherited]`

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.29.4.21 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()` `[static],[inherited]`

Definition at line 122 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 223 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 235 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.29.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const` `[inline],[inherited]`

Definition at line 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.29.4.33 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double *mean* = **DefaultPriorMeanI**, double *kappa* = **DefaultPriorIntensityKappa**) [static], [inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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* = **DefaultPriorBetaPos**) [static], [inherited]

Definition at line 99 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().

9.29.4.35 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (*IdxT size*, double *pos_sigma* = **DefaultPriorSigmaPos**) [static], [inherited]

Definition at line 92 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* = **DefaultPriorPSFSigmaAlpha**) [static], [inherited]

Definition at line 111 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 283 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 323 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 251 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_u`, `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 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.29.4.45 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline]`, `[inherited]`

Definition at line 275 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::DefaultPriorPixelMeanBG`, `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 220 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 267 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 247 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 227 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_I = -1) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::DefaultPriorMeanI, mappel::MCMCAdaptor1D::eta_I, 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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.29.4.56 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_) [inherited]`

Definition at line 165 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 156 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 117 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.29.4.59 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 244 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 264 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 303 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 67 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().

9.29.5.2 **const double mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95** [static], [inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.29.5.3 **const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"** [static], [inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.29.5.4 **const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10** [static], [inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.29.5.5 **const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300** [static], [inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.29.5.6 **const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0** [static], [inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.29.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.29.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.29.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.29.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.29.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.29.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.29.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.29.5.14 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.29.5.15 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.29.5.16 `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()`.

9.29.5.17 `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()`.

9.29.5.18 `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()`.

9.29.5.19 `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()`.

9.29.5.20 `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()`.

9.29.5.21 `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()`.

9.29.5.22 `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 69 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.29.5.23 `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 68 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.29.5.24 `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 155 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()`.

9.29.5.25 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

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

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.29.5.26 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 153 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()`.

9.29.5.27 `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()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

9.29.5.28 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 152 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()`.

9.29.5.29 `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()`, `MCMCAdaptor2D::D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

9.29.5.30 `ParamT mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 155 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()`.

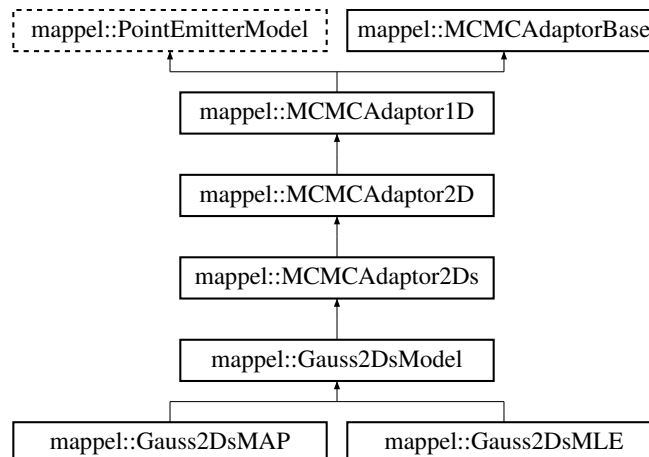
The documentation for this class was generated from the following files:

- [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=`DefaultPriorSigmaPos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` `size`, double `pos`↵
beta=`DefaultPriorBetaPos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double `mean`=`DefaultPriorMeanI`,
double `kappa`=`DefaultPriorIntensityKappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double `min_sigma`, double `max`↵
sigma, double `alpha`=`DefaultPriorPSFSigmaAlpha`)
- static void `set_rng_seed` (`RngSeedT` `seed`)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

Static Public Attributes

- static const std::string `DefaultEstimatorMethod` = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string `DefaultProfileBoundsEstimatorMethod` = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const `IdxT` `DefaultMCMCNumSamples` = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const `IdxT` `DefaultMCMCBurnin` = 10
Number of samples to throw away (burn-in) on initialization.
- static const `IdxT` `DefaultMCMCThin` = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.]
- static const double `DefaultConfidenceLevel` = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double `DefaultPriorBetaPos` = 3
- static const double `DefaultPriorSigmaPos` = 1
- static const double `DefaultPriorMeanI` = 300
- static const double `DefaultPriorMaxI` = INFINITY
- static const double `DefaultPriorIntensityKappa` = 2

- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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 [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 & 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 & o)` [protected]

Definition at line 22 of file MCMCAdaptor2Ds.cpp.

References `eta_sigma`.

9.30.3.4 `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds (MCMCAdaptor2Ds && 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 255 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 272 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 313 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 174 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 183 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 192 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 204 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 243 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.30.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`,`[inherited]`

Definition at line 263 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 239 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 231 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 219 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 215 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 167 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 255 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.30.4.18 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline],[inherited]`

Definition at line 207 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 211 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.30.4.20 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static],[inherited]`

Definition at line 127 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

9.30.4.21 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()` `[static],[inherited]`

Definition at line 122 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 223 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 235 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.30.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const [inline], [inherited]`

Definition at line 171 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 188 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 179 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 198 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 183 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 203 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 175 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 193 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.30.4.33 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double *mean* = DefaultPriorMeanI, double *kappa* = DefaultPriorIntensityKappa) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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 = DefaultPriorBetaPos*) [static],[inherited]

Definition at line 99 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 = DefaultPriorSigmaPos*) [static],[inherited]

Definition at line 92 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 = DefaultPriorPSFSigmaAlpha*) [static],[inherited]

Definition at line 111 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 283 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 323 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 251 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 271 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.30.4.45 **PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const** [inline],[inherited]

Definition at line 275 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::DefaultPriorPixelMeanBG, 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 220 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 267 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 247 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 227 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::DefaultPriorMeanI`, `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 233 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 259 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.30.4.56 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` [inherited]

Definition at line 165 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 156 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 117 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.30.4.59 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 244 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 264 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 303 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 67 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::DefaultConfidenceLevel = 0.95` [static],[inherited]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.30.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` [static],[inherited]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.30.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` [static],[inherited]

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.30.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` [static],[inherited]

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.30.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` [static],[inherited]

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.30.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.30.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.30.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.30.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

9.30.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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().

9.30.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.30.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.30.5.14 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.30.5.15 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), and mappel::Gauss2DsModel::initial_theta_estimate().

9.30.5.16 `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()`.

9.30.5.17 `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()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.30.5.18 `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()`.

9.30.5.19 `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()`.

9.30.5.20 `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()`.

9.30.5.21 `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()`.

9.30.5.22 `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()`.

9.30.5.23 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.30.5.24 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.30.5.25 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 155 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()`.

9.30.5.26 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.30.5.27 `IdxT mappel::PointEmitterModel::num_params` [protected],[inherited]

Definition at line 153 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()`.

9.30.5.28 `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::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

9.30.5.29 `CompositeDist mappel::PointEmitterModel::prior` [protected],[inherited]

Definition at line 152 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()`.

9.30.5.30 `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()`, `MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

9.30.5.31 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 155 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().

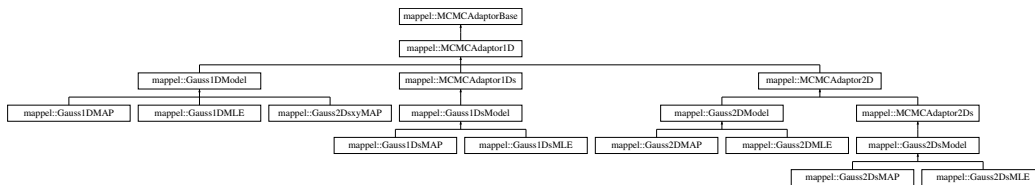
The documentation for this class was generated from the following files:

- [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_rllh

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::estimator::MLEDebugData Struct Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Public Member Functions

- [MLEData makeMLEData \(\)](#) const

Public Attributes

- [VecT theta](#)
Theta estimate.
- double [rllh](#)
RLLH at theta.
- [MatT obsI](#)
Observed Fisher information matrix at theta.
- [IdxT Nseq](#)
Number of points evaluated including theta_init and theta_mle.
- [MatT sequence](#)
Sequence of evaluated points including theta_init and theta_mle.
- [VecT sequence_rllh](#)
RLLH at each point in sequence.

9.35.1 Detailed Description

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.

9.35.2 Member Function Documentation

9.35.2.1 MLEData mappel::estimator::MLEDebugData::makeMLEData () const

Definition at line 18 of file estimator.cpp.

References [mappel::estimator::MLEData::obsI](#), [obsI](#), [mappel::estimator::MLEData::rllh](#), [rllh](#), [mappel::estimator::MLEData::theta](#), and [theta](#).

9.35.3 Member Data Documentation

9.35.3.1 IdxT mappel::estimator::MLEDebugData::Nseq

Number of points evaluated including theta_init and theta_mle.

Definition at line 55 of file estimator.h.

9.35.3.2 MatT mappel::estimator::MLEDebugData::obsI

Observed Fisher information matrix at theta.

Definition at line 54 of file estimator.h.

Referenced by mappel::estimator::Estimator< Model >::compute_estimate_debug(), mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::estimator::Estimator< Model >::estimate_max_debug(), mappel::estimator::CGaussMLE< Model >::get_debug_stats(), mappel::estimator::IterativeMaximizer< Model >::local_maximize(), mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize(), makeMLEData(), and mappel::estimator::subroutine::solve_profile_initial_step().

9.35.3.3 double mappel::estimator::MLEDebugData::rllh

RLLH at theta.

Definition at line 53 of file estimator.h.

Referenced by mappel::estimator::Estimator< Model >::compute_estimate_debug(), mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::estimator::Estimator< Model >::estimate_max_debug(), mappel::estimator::CGaussMLE< Model >::get_debug_stats(), mappel::estimator::IterativeMaximizer< Model >::local_maximize(), mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize(), and makeMLEData().

9.35.3.4 MatT mappel::estimator::MLEDebugData::sequence

Sequence of evaluated points including theta_init and theta_mle.

Definition at line 56 of file estimator.h.

Referenced by mappel::estimator::Estimator< Model >::compute_estimate_debug(), mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::estimator::CGaussMLE< Model >::get_debug_stats(), mappel::estimator::IterativeMaximizer< Model >::local_maximize(), and mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize().

9.35.3.5 VecT mappel::estimator::MLEDebugData::sequence_rllh

RLLH at each point in sequence.

Definition at line 57 of file estimator.h.

Referenced by mappel::estimator::Estimator< Model >::compute_estimate_debug(), mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::estimator::CGaussMLE< Model >::get_debug_stats(), mappel::estimator::IterativeMaximizer< Model >::local_maximize(), and mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize().

9.35.3.6 VecT mappel::estimator::MLEDebugData::theta

Theta estimate.

Definition at line 52 of file estimator.h.

Referenced by mappel::estimator::subroutine::compute_bound_scaling_vec(), mappel::estimator::Estimator< Model >::compute_estimate_debug(), mappel::estimator::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::estimator::Estimator< Model >::estimate_max_debug(), mappel::estimator::CGaussMLE< Model >::get_debug_stats(), mappel::estimator::IterativeMaximizer< Model >::local_maximize(), mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize(), and makeMLEData().

The documentation for this struct was generated from the following files:

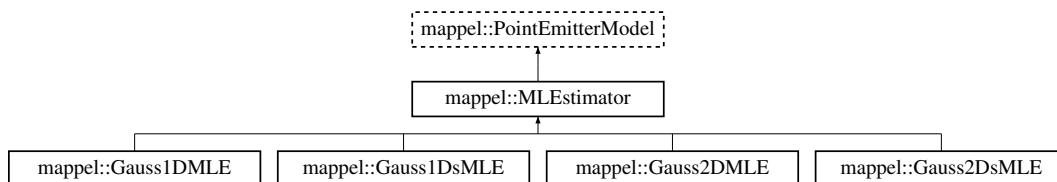
- [estimator.h](#)
- [estimator.cpp](#)

9.36 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_↵, double sigma=[DefaultPriorSigmaPos](#))
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_↵, double beta=[DefaultPriorBetaPos](#))
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[DefaultPriorMeanI](#), double kappa=[DefaultPriorIntensityKappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵, double sigma, double alpha=[DefaultPriorPSFSigmaAlpha](#))
- static void [set_rng_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()

Static Public Attributes

- static const std::string [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const IdxT [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const IdxT [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const IdxT [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested].
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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

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](#)

9.36.1 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.

9.36.2 Member Typedef Documentation

9.36.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [\[inherited\]](#)

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.36.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [\[inherited\]](#)

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

9.36.3 Constructor & Destructor Documentation

9.36.3.1 `mappel::MLEstimator::MLEstimator ()` [\[protected\]](#), [\[default\]](#)

9.36.3.2 `mappel::MLEstimator::MLEstimator (const MLEstimator & o)` [\[inline\]](#), [\[protected\]](#)

Definition at line 25 of file MLEstimator.h.

9.36.3.3 `mappel::MLEstimator::MLEstimator (MLEstimator && o)` [\[inline\]](#), [\[protected\]](#)

Definition at line 26 of file MLEstimator.h.

9.36.4 Member Function Documentation

9.36.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const` `[inherited]`

Definition at line 255 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

9.36.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` `[inherited]`

Definition at line 272 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()`.

9.36.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` `[inherited]`

Definition at line 313 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

9.36.4.4 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` `[inherited]`

Definition at line 174 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()`.

9.36.4.5 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` `[inherited]`

Definition at line 183 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

9.36.4.6 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` `[inherited]`

Definition at line 192 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()`.

9.36.4.7 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` `[inherited]`

Definition at line 204 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

9.36.4.8 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 243 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.36.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`, `[inherited]`

Definition at line 263 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.36.4.10 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 239 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.36.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 231 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.36.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline],`
`[inherited]`

Definition at line 219 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

9.36.4.13 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline], [inherited]`

Definition at line 215 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

9.36.4.14 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline], [inherited]`

Definition at line 167 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.36.4.15 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline], [inherited]`

Definition at line 255 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.36.4.16 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline], [inherited]`

Definition at line 207 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

9.36.4.17 `const CompositeDist & mappel::PointEmitterModel::get_prior () const` `[inline], [inherited]`

Definition at line 211 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.36.4.18 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static], [inherited]`

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.36.4.19 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 122 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.36.4.20 StatsT mappel::PointEmitterModel::get_stats () const [inherited]

Definition at line 132 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().

9.36.4.21 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 223 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().

9.36.4.22 bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline],[inherited]

Definition at line 235 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

9.36.4.23 PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const [inline],[inherited]

Definition at line 171 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().

9.36.4.24 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const [inherited]

Definition at line 188 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.36.4.25 **MatT** mappel::PointEmitterModel::make_param_mat () const [inline],[inherited]

Definition at line 179 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.36.4.26 **template<class FillT > MatT** mappel::PointEmitterModel::make_param_mat (FillT fill) const [inherited]

Definition at line 198 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.36.4.27 **CubeT** mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const [inline],[inherited]

Definition at line 183 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.36.4.28 **template<class FillT > CubeT** mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const [inherited]

Definition at line 203 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.36.4.29 **PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make_param_stack (IdxT n) const [inline],[inherited]

Definition at line 175 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta↔_stack().

9.36.4.30 **template<class FillT > PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const [inherited]

Definition at line 193 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

9.36.4.31 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double *mean* = **DefaultPriorMeanI**, double *kappa* = **DefaultPriorIntensityKappa**) [static],[inherited]

Definition at line 105 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::DefaultPriorMaxI.

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().

9.36.4.32 **prior_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make_prior_component_position_beta (IdxT *size*, double *pos_beta* = **DefaultPriorBetaPos**) [static],[inherited]

Definition at line 99 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().

9.36.4.33 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (IdxT *size*, double *pos_sigma* = **DefaultPriorSigmaPos**) [static],[inherited]

Definition at line 92 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().

9.36.4.34 **prior_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make_prior_component_sigma (double *min_sigma*, double *max_sigma*, double *alpha* = **DefaultPriorPSFSigmaAlpha**) [static],[inherited]

Definition at line 111 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().

9.36.4.35 `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=()`.

9.36.4.36 `MLEstimator& mappel::MLEstimator::operator= (MLEstimator && o)` `[inline],[protected]`

Definition at line 28 of file MLEstimator.h.

9.36.4.37 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const` `[inherited]`

Definition at line 283 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()`.

9.36.4.38 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const` `[inherited]`

Definition at line 323 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

9.36.4.39 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)` `[inline],[inherited]`

Definition at line 251 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.36.4.40 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng) const` `[inherited]`

Definition at line 271 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

9.36.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const` `[inline],[inherited]`

Definition at line 275 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

9.36.4.42 `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 220 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.36.4.43 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc)` `[inline]`,
`[inherited]`

Definition at line 267 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.36.4.44 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` `[inline]`,
`[inherited]`

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

References `mappel::PointEmitterModel::prior`.

9.36.4.45 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` `[inline]`, `[inherited]`

Definition at line 227 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

9.36.4.46 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 233 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()`.

9.36.4.47 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc)` `[inline]`, `[inherited]`

Definition at line 259 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

9.36.4.48 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_) [inherited]`

Definition at line 165 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()`.

9.36.4.49 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_) [inherited]`

Definition at line 156 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

9.36.4.50 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]`

Definition at line 117 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

9.36.4.51 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 244 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()`.

9.36.4.52 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const [inherited]`

Definition at line 264 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()`.

9.36.4.53 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const [inherited]`

Definition at line 303 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

9.36.5 Member Data Documentation

9.36.5.1 `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 67 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()`.

9.36.5.2 `const double mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95` `[static], [inherited]`

Default level at which to estimate confidence intervals must be in range (0,1).

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

9.36.5.3 `const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion"` `[static], [inherited]`

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file `PointEmitterModel.h`.

9.36.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` `[static], [inherited]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file `PointEmitterModel.h`.

9.36.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` `[static], [inherited]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file `PointEmitterModel.h`.

9.36.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` `[static], [inherited]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file `PointEmitterModel.h`.

9.36.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file `PointEmitterModel.h`.

9.36.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.36.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

9.36.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.36.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.36.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.36.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.36.5.14 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton" [static], [inherited]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.36.5.15 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static], [inherited]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.36.5.16 `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 69 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.36.5.17 `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 68 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

9.36.5.18 `ParamT mappel::PointEmitterModel::lbound [protected], [inherited]`

Definition at line 155 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()`.

9.36.5.19 `IdxT mappel::PointEmitterModel::num_hyperparams [protected], [inherited]`

Definition at line 154 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

9.36.5.20 `IdxT mappel::PointEmitterModel::num_params` [protected],[inherited]

Definition at line 153 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()`.

9.36.5.21 `CompositeDist mappel::PointEmitterModel::prior` [protected],[inherited]

Definition at line 152 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()`.

9.36.5.22 `ParamT mappel::PointEmitterModel::ubound` [protected],[inherited]

Definition at line 155 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:

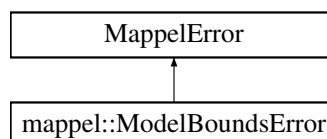
- [MLEstimator.h](#)

9.37 `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)

9.37.1 Detailed Description

Access outside the model bounds is attempted.

Definition at line 90 of file util.h.

9.37.2 Constructor & Destructor Documentation

9.37.2.1 mappel::ModelBoundsError::ModelBoundsError (std::string *message*) [inline]

Definition at line 92 of file util.h.

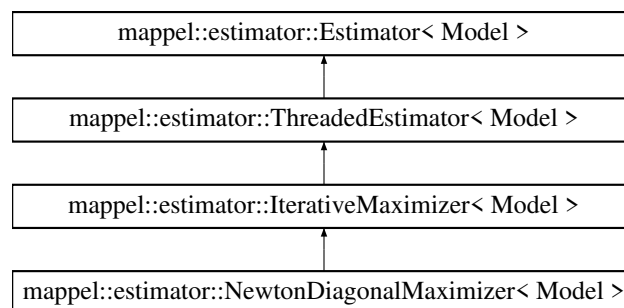
The documentation for this struct was generated from the following file:

- [util.h](#)

9.38 mappel::estimator::NewtonDiagonalMaximizer< Model > 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_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, 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

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, [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_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::NewtonDiagonalMaximizer< Model >
```

Definition at line 569 of file estimator.h.

9.38.2 Member Typedef Documentation

9.38.2.1 `template<class Model > using mappel::estimator::NewtonDiagonalMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 572 of file estimator.h.

9.38.3 Constructor & Destructor Documentation

9.38.3.1 `template<class Model > mappel::estimator::NewtonDiagonalMaximizer< Model >::NewtonDiagonalMaximizer (const Model & model, int max_iterations = IterativeMaximizer<Model>::DefaultIterations)`
[inline]

Definition at line 574 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>	MLEData 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>	MLEData 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>	MLEData 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>	MLEDebugData 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>	MLEData 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>	MLEData 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_idxxs`, `mappel::estimator::ProfileBoundsDataStack::estimated_idxxs`, `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::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()`.

9.38.4.21 `template<class Model > double mappel::estimator::Estimator< Model >::estimate_profile_max (const ModelDataT< Model > & data, const IdxVecT & fixed_idxxs, 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_idxxs`, `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 276 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 443 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 445 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 444 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 442 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::NewtonDiagonalMaximizer< Model >::name () const`
[inline],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 577 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 462 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 463 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 461 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 458 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 459 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 430 of file estimator.h.

9.38.5.7 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 301 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 477 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 464 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 468 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 326 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 456 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 466 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 296 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 328 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 299 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 327 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 472 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 474 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 473 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 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()`.

9.38.5.22 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected], [inherited]`

Definition at line 300 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()`.

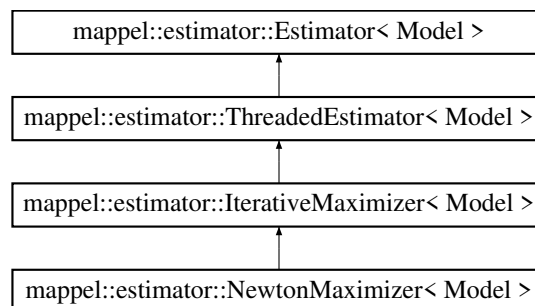
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

9.39 mappel::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_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 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.

9.39.1 Detailed Description

```
template<class Model>
class mappel::estimator::NewtonMaximizer< Model >
```

Definition at line 584 of file estimator.h.

9.39.2 Member Typedef Documentation

9.39.2.1 `template<class Model> using mappel::estimator::NewtonMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 587 of file estimator.h.

9.39.3 Constructor & Destructor Documentation

9.39.3.1 `template<class Model> mappel::estimator::NewtonMaximizer< Model >::NewtonMaximizer (const Model & model, int max_iterations = IterativeMaximizer<Model>::DefaultIterations) [inline]`

Definition at line 589 of file estimator.h.

9.39.4 Member Function Documentation

9.39.4.1 `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()`.

9.39.4.2 `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`.

9.39.4.3 `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()`.

9.39.4.4 `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()`.

9.39.4.5 `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`.

9.39.4.6 `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`.

9.39.4.7 `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()`.

9.39.4.8 `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()`.

9.39.4.9 `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()`.

9.39.4.10 `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>	MLEData 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().

9.39.4.11 `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>	MLEData 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().

9.39.4.12 `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>	MLEData 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.

9.39.4.13 **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>	MLEDebugData recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.39.4.14 **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>	MLEData 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.

9.39.4.15 `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>	MLEData 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](#).

9.39.4.16 `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.

9.39.4.17 `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().

9.39.4.18 `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().

9.39.4.19 `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().

9.39.4.20 `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\(\)](#).

9.39.4.21 `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\(\)](#).

9.39.4.22 `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\(\)](#).

9.39.4.23 `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.

9.39.4.24 `template<class Model> IdxVecT mappel::estimator::Estimator< Model>::get_exit_counts () const`
`[inline], [inherited]`

Run statistics.

Definition at line 276 of file `estimator.h`.

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

9.39.4.25 `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`.

9.39.4.26 `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()`.

9.39.4.27 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_backtracks () const`
`[inline], [inherited]`

Definition at line 443 of file estimator.h.

9.39.4.28 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_der_evals () const`
`[inline], [inherited]`

Definition at line 445 of file estimator.h.

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

9.39.4.29 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_fun_evals () const`
`[inline], [inherited]`

Definition at line 444 of file estimator.h.

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

9.39.4.30 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::get_total_iterations () const`
`[inline], [inherited]`

Definition at line 442 of file estimator.h.

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

9.39.4.31 `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()`.

9.39.4.32 `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()`.

9.39.4.33 `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()`.

9.39.4.34 `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()`.

9.39.4.35 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_backtracks ()`
[inherited]

9.39.4.36 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_der_evals ()`
[inherited]

9.39.4.37 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_fun_evals ()`
[inherited]

9.39.4.38 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_iterations ()`
[inherited]

9.39.4.39 `template<class Model> std::string mappel::estimator::NewtonMaximizer< Model >::name () const`
[inline],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 592 of file estimator.h.

9.39.4.40 `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\(\)](#).

9.39.4.41 `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()`.

9.39.4.42 `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()`.

9.39.4.43 `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()`.

9.39.4.44 `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().

9.39.5 Member Data Documentation

9.39.5.1 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_max_ratio = 0.50` [static], [protected], [inherited]

Definition at line 462 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.39.5.2 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio = 1e-3` [static], [protected], [inherited]

Definition at line 463 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::backtrack(), and mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.39.5.3 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_ratio = 0.05` [static], [protected], [inherited]

Definition at line 461 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.39.5.4 `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 458 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio(), and mappel::estimator::IterativeMaximizer< Model >::get_stats().

9.39.5.5 `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 459 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()`.

9.39.5.6 `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::DefaultIterations = 100` [static], [inherited]

Definition at line 430 of file estimator.h.

9.39.5.7 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 301 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()`.

9.39.5.8 `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 477 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.39.5.9 `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::max_backtracks = 8` [static], [protected], [inherited]

Definition at line 464 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()`.

9.39.5.10 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::max_iterations` [protected], [inherited]

Definition at line 468 of file estimator.h.

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

9.39.5.11 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::max_threads`
`[protected], [inherited]`

Definition at line 326 of file estimator.h.

9.39.5.12 `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 456 of file estimator.h.

9.39.5.13 `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 466 of file estimator.h.

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

9.39.5.14 `template<class Model > const Model& mappel::estimator::Estimator< Model >::model` `[protected],`
`[inherited]`

Definition at line 296 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()`.

9.39.5.15 `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` `[protected],`
`[inherited]`

Definition at line 328 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()`.

9.39.5.16 `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [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::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()`.

9.39.5.17 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 327 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()`.

9.39.5.18 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_backtracks = 0` [protected], [inherited]

Definition at line 472 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()`.

9.39.5.19 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_der_evals = 0` [protected], [inherited]

Definition at line 474 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()`.

9.39.5.20 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_fun_evals = 0` [protected], [inherited]

Definition at line 473 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()`.

9.39.5.21 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_iterations = 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()`.

9.39.5.22 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected], [inherited]`

Definition at line 300 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:

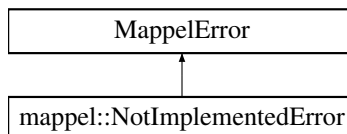
- [estimator.h](#)
- [estimator_impl.h](#)

9.40 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)

9.40.1 Detailed Description

Feature not yet implemented.

Definition at line 111 of file util.h.

9.40.2 Constructor & Destructor Documentation

9.40.2.1 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:

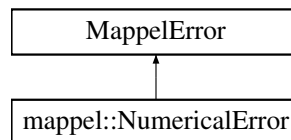
- [util.h](#)

9.41 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)

9.41.1 Detailed Description

Expected numerical condition does not hold.

Definition at line 97 of file util.h.

9.41.2 Constructor & Destructor Documentation

9.41.2.1 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](#)

9.42 `omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>` Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/OMPExcceptionCatcher/OMPExcceptionCatcher.h>
```

Public Member Functions

- [OMPExcceptionCatcher](#) ()
- [OMPExcceptionCatcher](#) ([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)

9.42.1 Detailed Description

```
template<class _dummy = void>  
class omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>
```

Implementation of [OMPExcceptionCatcher](#)

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 `OMPExcceptionCatcher.h`.

9.42.2 Constructor & Destructor Documentation

9.42.2.1 `template<class _dummy = void> omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::OMPExcceptionCatcher () [inline]`

Construct a new [OMPExcceptionCatcher](#) using the `GlobalDefaultStrategy`

Definition at line 55 of file `OMPExcceptionCatcher.h`.

9.42.2.2 `template<class _dummy = void> omp_exception_catcher::impl_::OMPExcceptionCatcher<_dummy>::OMPExcceptionCatcher (Strategy strategy_) [inline]`

Construct a new [OMPExcceptionCatcher](#) using the given strategy

Definition at line 59 of file `OMPExcceptionCatcher.h`.

9.42.3 Member Function Documentation

9.42.3.1 `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()`.

9.42.3.2 `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()`.

9.42.3.3 `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:

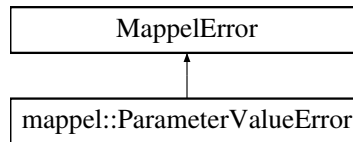
- [OMPEXceptionCatcher.h](#)

9.43 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)

9.43.1 Detailed Description

Parameter value is not valid.

Definition at line 69 of file util.h.

9.43.2 Constructor & Destructor Documentation

9.43.2.1 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:

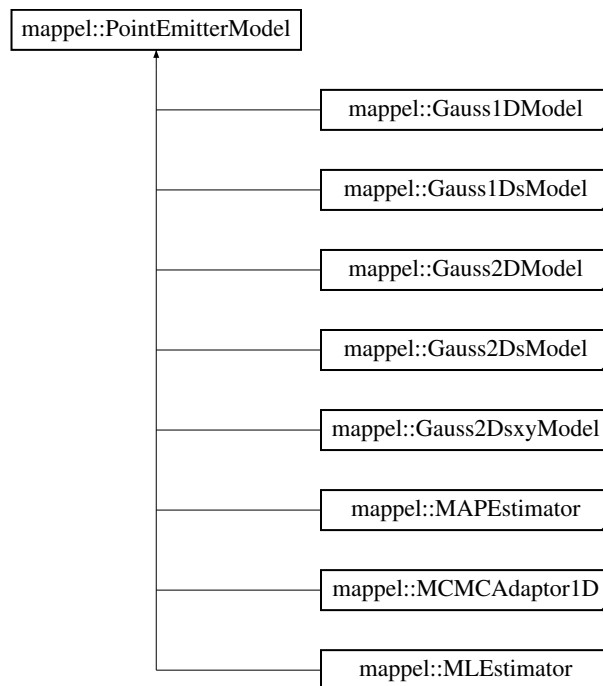
- [util.h](#)

9.44 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_↵, sigma=DefaultPriorSigmaPos)`
- `static prior_hessian::ScaledSymmetricBetaDist make_prior_component_position_beta (IdxT size, double pos_↵, beta=DefaultPriorBetaPos)`
- `static prior_hessian::TruncatedGammaDist make_prior_component_intensity (double mean=DefaultPriorMeanI, double kappa=DefaultPriorIntensityKappa)`
- `static prior_hessian::TruncatedParetoDist make_prior_component_sigma (double min_sigma, double max_↵, sigma, double alpha=DefaultPriorPSFSigmaAlpha)`
- `static void set_rng_seed (RngSeedT seed)`
- `static ParallelRngManagerT & get_rng_manager ()`
- `static ParallelRngGeneratorT & get_rng_generator ()`

Static Public Attributes

- static const std::string [DefaultEstimatorMethod](#) = "TrustRegion"
Default optimization method for MLE/MAP estimation.
- static const std::string [DefaultProfileBoundsEstimatorMethod](#) = "Newton"
Default optimization method for profile bounds optimizations.
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const [IdxT](#) [DefaultMCMCNumSamples](#) = 300
Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)
- static const [IdxT](#) [DefaultMCMCBurnin](#) = 10
Number of samples to throw away (burn-in) on initialization.
- static const [IdxT](#) [DefaultMCMCThin](#) = 0
Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].
- static const double [DefaultConfidenceLevel](#) = 0.95
Default level at which to estimate confidence intervals must be in range (0,1).
- static const double [DefaultPriorBetaPos](#) = 3
- static const double [DefaultPriorSigmaPos](#) = 1
- static const double [DefaultPriorMeanI](#) = 300
- static const double [DefaultPriorMaxI](#) = INFINITY
- static const double [DefaultPriorIntensityKappa](#) = 2
- static const double [DefaultPriorPixelMeanBG](#) = 4
- static const double [DefaultPriorPSFSigmaAlpha](#) = 2
- 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

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](#)

9.44.1 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.

9.44.2 Member Typedef Documentation

9.44.2.1 using mappel::PointEmitterModel::ParamT = arma::vec

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

9.44.2.2 using mappel::PointEmitterModel::ParamVecT = arma::mat

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

9.44.3 Constructor & Destructor Documentation

9.44.3.1 mappel::PointEmitterModel::PointEmitterModel () [protected]

Definition at line 38 of file PointEmitterModel.cpp.

9.44.3.2 mappel::PointEmitterModel::PointEmitterModel (const CompositeDist & prior_) [explicit], [protected]

Definition at line 50 of file PointEmitterModel.cpp.

9.44.3.3 mappel::PointEmitterModel::PointEmitterModel (CompositeDist && prior_) [explicit], [protected]

Definition at line 44 of file PointEmitterModel.cpp.

9.44.3.4 mappel::PointEmitterModel::PointEmitterModel (const PointEmitterModel & o) [protected]

Definition at line 56 of file PointEmitterModel.cpp.

References prior.

9.44.3.5 mappel::PointEmitterModel::PointEmitterModel (PointEmitterModel && o) [protected]

Definition at line 62 of file PointEmitterModel.cpp.

9.44.4 Member Function Documentation

9.44.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const

Definition at line 255 of file PointEmitterModel.cpp.

References check_param_shape(), lbound, num_params, and ubound.

9.44.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const

Definition at line 272 of file PointEmitterModel.cpp.

References check_param_shape(), lbound, num_params, and ubound.

Referenced by bounded_theta_stack().

9.44.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const

Definition at line 313 of file PointEmitterModel.cpp.

References bounded_theta(), check_param_shape(), and make_param_stack().

9.44.4.4 void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const

Definition at line 174 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().

9.44.4.5 void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const

Definition at line 183 of file PointEmitterModel.cpp.

References num_params.

9.44.4.6 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const`

Definition at line 192 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()`.

9.44.4.7 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const`

Definition at line 204 of file PointEmitterModel.cpp.

References `global_max_psf_sigma`, and `global_min_psf_sigma`.

9.44.4.8 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`

Definition at line 243 of file PointEmitterModel.h.

References `prior`.

9.44.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`

Definition at line 263 of file PointEmitterModel.h.

References `prior`.

9.44.4.10 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`

Definition at line 239 of file PointEmitterModel.h.

References `prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.44.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`

Definition at line 231 of file PointEmitterModel.h.

References `prior`.

9.44.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`

Definition at line 219 of file PointEmitterModel.h.

References `lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

9.44.4.13 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const [inline]`

Definition at line 215 of file PointEmitterModel.h.

References num_hyperparams.

9.44.4.14 `IdxT mappel::PointEmitterModel::get_num_params () const [inline]`

Definition at line 167 of file PointEmitterModel.h.

References num_params.

9.44.4.15 `StringVecT mappel::PointEmitterModel::get_param_names () const [inline]`

Definition at line 255 of file PointEmitterModel.h.

References prior.

9.44.4.16 `CompositeDist & mappel::PointEmitterModel::get_prior () [inline]`

Definition at line 207 of file PointEmitterModel.h.

References prior.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

9.44.4.17 `const CompositeDist & mappel::PointEmitterModel::get_prior () const [inline]`

Definition at line 211 of file PointEmitterModel.h.

References prior.

9.44.4.18 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static]`

Definition at line 127 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.44.4.19 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static]`

Definition at line 122 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.44.4.20 StatsT mappel::PointEmitterModel::get_stats () const

Definition at line 132 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().

9.44.4.21 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline]

Definition at line 223 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().

9.44.4.22 bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline]

Definition at line 235 of file PointEmitterModel.h.

References prior.

9.44.4.23 PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const [inline]

Definition at line 171 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().

9.44.4.24 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const

Definition at line 188 of file PointEmitterModel.h.

References num_params.

9.44.4.25 MatT mappel::PointEmitterModel::make_param_mat () const [inline]

Definition at line 179 of file PointEmitterModel.h.

References num_params.

9.44.4.26 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const`

Definition at line 198 of file PointEmitterModel.h.

References num_params.

9.44.4.27 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const [inline]`

Definition at line 183 of file PointEmitterModel.h.

References num_params.

9.44.4.28 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`

Definition at line 203 of file PointEmitterModel.h.

References num_params.

9.44.4.29 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const [inline]`

Definition at line 175 of file PointEmitterModel.h.

References num_params.

Referenced by bounded_theta_stack(), and reflected_theta_stack().

9.44.4.30 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const`

Definition at line 193 of file PointEmitterModel.h.

References num_params.

9.44.4.31 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = DefaultPriorMeanI, double kappa = DefaultPriorIntensityKappa) [static]`

Definition at line 105 of file PointEmitterModel.cpp.

References DefaultPriorMaxI.

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().

9.44.4.32 **prior_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make_prior_component_position_beta (*IdxT size*, *double pos_beta = DefaultPriorBetaPos*) [static]

Definition at line 99 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().

9.44.4.33 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (*IdxT size*, *double pos_sigma = DefaultPriorSigmaPos*) [static]

Definition at line 92 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::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

9.44.4.34 **prior_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make_prior_component_sigma (*double min_sigma*, *double max_sigma*, *double alpha = DefaultPriorPSFSigmaAlpha*) [static]

Definition at line 111 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().

9.44.4.35 **PointEmitterModel & mappel::PointEmitterModel::operator= (const PointEmitterModel & o)** [protected]

Definition at line 68 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=().

9.44.4.36 **PointEmitterModel & mappel::PointEmitterModel::operator= (PointEmitterModel && o)** [protected]

Definition at line 75 of file PointEmitterModel.cpp.

References lbound, num_hyperparams, num_params, prior, and ubound.

9.44.4.37 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const`

Definition at line 283 of file `PointEmitterModel.cpp`.

References `check_param_shape()`, `lbound`, `num_params`, and `ubound`.

Referenced by `reflected_theta_stack()`.

9.44.4.38 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const`

Definition at line 323 of file `PointEmitterModel.cpp`.

References `check_param_shape()`, `make_param_stack()`, and `reflected_theta()`.

9.44.4.39 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name) [inline]`

Definition at line 251 of file `PointEmitterModel.h`.

References `prior`.

9.44.4.40 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng) const`

Definition at line 271 of file `PointEmitterModel.h`.

References `prior`.

9.44.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () const [inline]`

Definition at line 275 of file `PointEmitterModel.h`.

References `prior`, and `mappel::rng_manager`.

9.44.4.42 `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 220 of file `PointEmitterModel.cpp`.

References `bounds_epsilon`, `lbound`, `num_params`, `prior`, and `ubound`.

9.44.4.43 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc) [inline]`

Definition at line 267 of file `PointEmitterModel.h`.

References `prior`.

9.44.4.44 void mappel::PointEmitterModel::set_hyperparam_value (const std::string & *name*, double *value*) [inline]

Definition at line 247 of file PointEmitterModel.h.

References prior.

9.44.4.45 void mappel::PointEmitterModel::set_hyperparams (const VecT & *hyperparams*) [inline]

Definition at line 227 of file PointEmitterModel.h.

References prior.

Referenced by mappel::Gauss2DModel::set_hyperparams(), and mappel::Gauss2DsModel::set_hyperparams().

9.44.4.46 void mappel::PointEmitterModel::set_lbound (const ParamT & *lbound*)

Definition at line 233 of file PointEmitterModel.cpp.

References bounds_epsilon, lbound, num_params, prior, and ubound.

Referenced by mappel::Gauss1DsModel::set_min_sigma().

9.44.4.47 void mappel::PointEmitterModel::set_param_names (const StringVecT & *desc*) [inline]

Definition at line 259 of file PointEmitterModel.h.

References prior.

9.44.4.48 void mappel::PointEmitterModel::set_prior (CompositeDist && *prior_*)

Definition at line 165 of file PointEmitterModel.cpp.

References lbound, num_hyperparams, num_params, prior, and ubound.

Referenced by mappel::Gauss2DModel::set_prior(), and mappel::Gauss2DsModel::set_prior().

9.44.4.49 void mappel::PointEmitterModel::set_prior (const CompositeDist & *prior_*)

Definition at line 156 of file PointEmitterModel.cpp.

References lbound, num_hyperparams, num_params, prior, and ubound.

9.44.4.50 void mappel::PointEmitterModel::set_rng_seed (RngSeedT *seed*) [static]

Definition at line 117 of file PointEmitterModel.cpp.

References mappel::rng_manager.

9.44.4.51 void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)

Definition at line 244 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().

9.44.4.52 bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const

Definition at line 264 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().

9.44.4.53 BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const

Definition at line 303 of file PointEmitterModel.cpp.

References check_param_shape(), and theta_in_bounds().

9.44.5 Member Data Documentation

9.44.5.1 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 67 of file PointEmitterModel.h.

Referenced by set_bounds(), set_lbound(), mappel::Gauss2DsModel::set_max_sigma_ratio(), and set_ubound().

9.44.5.2 const double mappel::PointEmitterModel::DefaultConfidenceLevel = 0.95 [static]

Default level at which to estimate confidence intervals must be in range (0,1).

Definition at line 57 of file PointEmitterModel.h.

9.44.5.3 const std::string mappel::PointEmitterModel::DefaultEstimatorMethod = "TrustRegion" [static]

Default optimization method for MLE/MAP estimation.

Definition at line 51 of file PointEmitterModel.h.

9.44.5.4 `const IdxT mappel::PointEmitterModel::DefaultMCMCBurnin = 10` `[static]`

Number of samples to throw away (burn-in) on initialization.

Definition at line 55 of file PointEmitterModel.h.

9.44.5.5 `const IdxT mappel::PointEmitterModel::DefaultMCMCNumSamples = 300` `[static]`

Number of final samples to use in estimation of posterior properties (mean, credible interval, cov, etc.)

Definition at line 54 of file PointEmitterModel.h.

9.44.5.6 `const IdxT mappel::PointEmitterModel::DefaultMCMCThin = 0` `[static]`

Keep every # samples. [Value of 0 indicates use the model default. This is suggested.].

Definition at line 56 of file PointEmitterModel.h.

9.44.5.7 `const double mappel::PointEmitterModel::DefaultPriorBetaPos = 3` `[static]`

Default position parameter in symmetric beta-distributions

Definition at line 59 of file PointEmitterModel.h.

9.44.5.8 `const double mappel::PointEmitterModel::DefaultPriorIntensityKappa = 2` `[static]`

Default shape for intensity gamma distributions

Definition at line 63 of file PointEmitterModel.h.

9.44.5.9 `const double mappel::PointEmitterModel::DefaultPriorMaxI = INFINITY` `[static]`

Default maximum emitter intensity

Definition at line 62 of file PointEmitterModel.h.

Referenced by `make_prior_component_intensity()`.

9.44.5.10 `const double mappel::PointEmitterModel::DefaultPriorMeanI = 300` `[static]`

Default emitter intensity mean

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

9.44.5.11 `const double mappel::PointEmitterModel::DefaultPriorPixelMeanBG = 4` `[static]`

Default per-pixel mean background counts

Definition at line 64 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()`.

9.44.5.12 `const double mappel::PointEmitterModel::DefaultPriorPSFSigmaAlpha = 2` `[static]`

Default per-pixel background gamma distribution shape

Definition at line 65 of file PointEmitterModel.h.

9.44.5.13 `const double mappel::PointEmitterModel::DefaultPriorSigmaPos = 1` `[static]`

Default position parameter in symmetric beta-distributions

Definition at line 60 of file PointEmitterModel.h.

9.44.5.14 `const std::string mappel::PointEmitterModel::DefaultProfileBoundsEstimatorMethod = "Newton"` `[static]`

Default optimization method for profile bounds optimizations.

Definition at line 52 of file PointEmitterModel.h.

9.44.5.15 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static]`

Estimator name to use in 1D separable initializations

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

9.44.5.16 `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 69 of file PointEmitterModel.h.

Referenced by `check_psf_sigma()`.

9.44.5.17 `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 68 of file `PointEmitterModel.h`.

Referenced by `check_psf_sigma()`.

9.44.5.18 `ParamT mappel::PointEmitterModel::lbound` [protected]

Definition at line 155 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()`.

9.44.5.19 `IdxT mappel::PointEmitterModel::num_hyperparams` [protected]

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

Referenced by `get_num_hyperparams()`, `get_stats()`, `operator=()`, and `set_prior()`.

9.44.5.20 `IdxT mappel::PointEmitterModel::num_params` [protected]

Definition at line 153 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()`.

9.44.5.21 `CompositeDist mappel::PointEmitterModel::prior` [protected]

Definition at line 152 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()`.

9.44.5.22 `ParamT mappel::PointEmitterModel::ubound` [protected]

Definition at line 155 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:

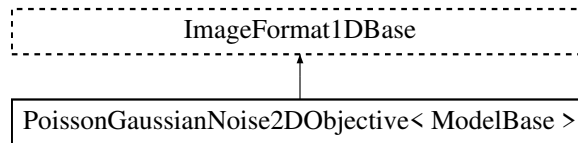
- [PointEmitterModel.h](#)
- [PointEmitterModel.cpp](#)

9.45 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](#)

9.45.1 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.

9.45.2 Member Typedef Documentation

9.45.2.1 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordIdxT = uint32_t`

Definition at line 26 of file PoissonGaussianNoise2DObjective.h.

9.45.2.2 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordStackT = arma::mat<uint32_t>`

Definition at line 28 of file PoissonGaussianNoise2DObjective.h.

9.45.2.3 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordT = arma::vec<uint32_t>`

Definition at line 27 of file PoissonGaussianNoise2DObjective.h.

9.45.2.4 `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.

9.45.2.5 `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.

9.45.3 Constructor & Destructor Documentation

9.45.3.1 `template<typename ModelBase > PoissonGaussianNoise2DObjective< ModelBase >::PoissonGaussianNoise2DObjective (const ImageSizeVecT & size, const ImageT & sensor_gain_map, const ImageT & sensor_bg_map)`

9.45.4 Member Data Documentation

9.45.4.1 `template<typename ModelBase > const std::vector< std::string > PoissonGaussianNoise2DObjective< ModelBase >::estimator_names [static]`

Definition at line 25 of file PoissonGaussianNoise2DObjective.h.

9.45.4.2 `template<typename ModelBase > ImageT PoissonGaussianNoise2DObjective< ModelBase >::sensor_bg_map`

Definition at line 34 of file `PoissonGaussianNoise2DObjective.h`.

9.45.4.3 `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:

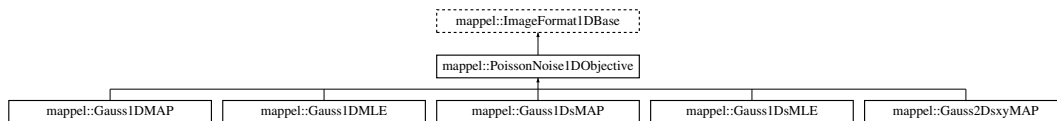
- [PoissonGaussianNoise2DObjective.h](#)
- [PoissonGaussianNoise2DObjective.cpp](#)

9.46 `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](#)

9.46.1 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.

9.46.2 Member Typedef Documentation

9.46.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

9.46.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

9.46.2.3 `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.

9.46.2.4 `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.

9.46.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

9.46.2.6 `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.

9.46.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
[inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

9.46.2.8 `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.

9.46.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
[inherited]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

9.46.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

9.46.2.11 `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.

9.46.2.12 `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.

9.46.3 Constructor & Destructor Documentation

9.46.3.1 `mappel::PoissonNoise1DObjective::PoissonNoise1DObjective ()` [protected]

Definition at line 14 of file PoissonNoise1DObjective.cpp.

9.46.3.2 `mappel::PoissonNoise1DObjective::PoissonNoise1DObjective (const PoissonNoise1DObjective & o)`
`[protected]`

Definition at line 18 of file PoissonNoise1DObjective.cpp.

9.46.3.3 `mappel::PoissonNoise1DObjective::PoissonNoise1DObjective (PoissonNoise1DObjective && o)`
`[protected]`

Definition at line 22 of file PoissonNoise1DObjective.cpp.

9.46.4 Member Function Documentation

9.46.4.1 `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`.

9.46.4.2 `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`.

9.46.4.3 `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()`.

9.46.4.4 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack,`
`ImageCoordT n) const` `[inline],[inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

9.46.4.5 `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()`.

9.46.4.6 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().

9.46.4.7 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT idx) const [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

9.46.4.8 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

9.46.4.9 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().

9.46.4.10 ImageFormat1DBase::ImageT ImageFormat1DBase::make_image () const [inline],[inherited]

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

9.46.4.11 ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack (ImageCoordT n) const [inline],[inherited]

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

9.46.4.12 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=().

9.46.4.13 `PoissonNoise1DObjective & mappel::PoissonNoise1DObjective::operator= (PoissonNoise1DObjective && o)`
`[protected]`

Definition at line 31 of file PoissonNoise1DObjective.cpp.

9.46.4.14 `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.

9.46.4.15 `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::Gauss2DModel::set_size()`.

9.46.4.16 `void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & sz)` `[inline], [inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

9.46.5 Member Data Documentation

9.46.5.1 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names` `[static]`

Definition at line 24 of file PoissonNoise1DObjective.h.

9.46.5.2 `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()`.

9.46.5.3 `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()`.

9.46.5.4 `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()`.

9.46.5.5 `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:

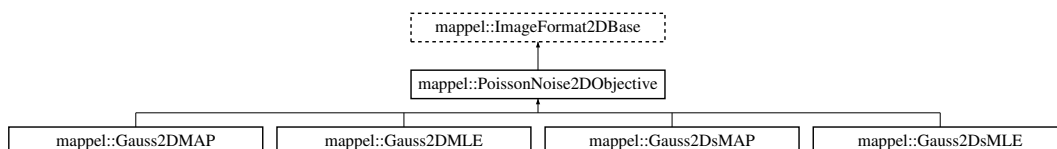
- [PoissonNoise1DObjective.h](#)
- [PoissonNoise1DObjective.cpp](#)

9.47 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

9.47.1 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.

9.47.2 Member Typedef Documentation

9.47.2.1 using mappel::ImageFormat2DBase::ImageCoordT = uint32_t [inherited]

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

9.47.2.2 using mappel::ImageFormat2DBase::ImagePixelT = double [inherited]

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

9.47.2.3 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.

9.47.2.4 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.

9.47.2.5 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
`[inherited]`

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

9.47.2.6 `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.

9.47.2.7 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

9.47.2.8 `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.

9.47.2.9 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

9.47.2.10 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

9.47.2.11 `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.

9.47.2.12 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.

9.47.3 Constructor & Destructor Documentation

9.47.3.1 mappel::PoissonNoise2DObjective::PoissonNoise2DObjective () [protected]

Definition at line 15 of file PoissonNoise2DObjective.cpp.

9.47.3.2 mappel::PoissonNoise2DObjective::PoissonNoise2DObjective (const PoissonNoise2DObjective & o) [protected]

Definition at line 19 of file PoissonNoise2DObjective.cpp.

9.47.3.3 mappel::PoissonNoise2DObjective::PoissonNoise2DObjective (PoissonNoise2DObjective && o) [protected]

Definition at line 23 of file PoissonNoise2DObjective.cpp.

9.47.4 Member Function Documentation

9.47.4.1 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.

9.47.4.2 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.

9.47.4.3 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().

9.47.4.4 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const [inline],[inherited]

Definition at line 106 of file ImageFormat2DBase.h.

9.47.4.5 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().

9.47.4.6 const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const [inline],[inherited]

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

9.47.4.7 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT idx) const [inherited]

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

9.47.4.8 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

9.47.4.9 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().

9.47.4.10 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const [inline],[inherited]

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

9.47.4.11 `ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT n) const`
`[inline], [inherited]`

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

9.47.4.12 `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=().

9.47.4.13 `PoissonNoise2DObjective & mappel::PoissonNoise2DObjective::operator= (PoissonNoise2DObjective && o)`
`[protected]`

Definition at line 32 of file PoissonNoise2DObjective.cpp.

9.47.4.14 `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.

9.47.4.15 `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().

9.47.5 Member Data Documentation

9.47.5.1 `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` `[static]`

Definition at line 23 of file PoissonNoise2DObjective.h.

9.47.5.2 `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().

9.47.5.3 `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()`.

9.47.5.4 `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()`.

9.47.5.5 `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::Gauss2Ds`
`Model::Stencil::Stencil()`, `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2Ds`
`Model::update_internal_1Dsum_estimators()`.

The documentation for this class was generated from the following files:

- [PoissonNoise2DObjective.h](#)
- [PoissonNoise2DObjective.cpp](#)

9.48 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_idxs](#)
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 ($-\chi^2_{inv}(confidence, 1)/2$)
- [IdxT](#) [Nparams_est](#) =0
number of parameters estimated =estimated_param_idxs.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

9.48.1 Detailed Description

Data related to a profile bounds estimation for a single image Includes both controlling (input) parameters as well as reporting (ouput) parameters to give output parameters context.

Definition at line 92 of file estimator.h.

9.48.2 Member Function Documentation

9.48.2.1 void mappel::estimator::ProfileBoundsData::initialize_arrays ([IdxT](#) Nparams)

Definition at line 27 of file estimator.cpp.

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\(\)](#).

9.48.3 Member Data Documentation

9.48.3.1 `double mappel::estimator::ProfileBoundsData::confidence = 1`

Confidence level. If invalid, use default value.

Definition at line 96 of file estimator.h.

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

9.48.3.2 `IdxVecT mappel::estimator::ProfileBoundsData::estimated_idxxs`

List of indexs for computed parameters. Empty to compute all parameters.

Definition at line 95 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()`.

9.48.3.3 `MLEData mappel::estimator::ProfileBoundsData::mle`

Theta maximum-likelihood estimate, rllh, and Obsl.

Definition at line 97 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()`.

9.48.3.4 `IdxT mappel::estimator::ProfileBoundsData::Nparams_est = 0`

number of parameters estimated = `estimated_param_idxxs.n_elem`.

Definition at line 101 of file estimator.h.

Referenced by `mappel::estimator::Estimator< Model >::estimate_profile_bounds()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_parallel()`.

9.48.3.5 `VecT mappel::estimator::ProfileBoundsData::profile_lb`

size:[`Nparams_est`] Lower bound estimated at 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()`, and `mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack()`.

9.48.3.6 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 104 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound(), mappel::estimator::Estimator< Model >::estimate_profile_bounds(), and mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.48.3.7 VecT mappel::estimator::ProfileBoundsData::profile_points_lb_rllh

size:[Nparams_est] RLLH at each of the profile_points_lb

Definition at line 106 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound(), mappel::estimator::Estimator< Model >::estimate_profile_bounds(), and mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.48.3.8 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 105 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound(), mappel::estimator::Estimator< Model >::estimate_profile_bounds(), and mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.48.3.9 VecT mappel::estimator::ProfileBoundsData::profile_points_ub_rllh

size:[Nparams_est] RLLH at each of the profile_points_lb

Definition at line 107 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::compute_profile_bound(), and mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.48.3.10 VecT mappel::estimator::ProfileBoundsData::profile_ub

size:[Nparams_est] Upper bound estimated at 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(), and mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.48.3.11 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 100 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](#)

9.49 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_idxxs](#)
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_idxxs.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

9.49.1 Detailed Description

Data related to a profile bounds estimation for a stack of images Includes both controlling (input) parameters as well as reporting (output) parameters to give output parameters context.

Definition at line 136 of file estimator.h.

9.49.2 Member Function Documentation

9.49.2.1 void mappel::estimator::ProfileBoundsDataStack::initialize_arrays (*IdxT Nparams*)

Definition at line 38 of file estimator.cpp.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3 Member Data Documentation

9.49.3.1 double mappel::estimator::ProfileBoundsDataStack::confidence =-1

Confidence level. If invalid, use default value.

Definition at line 139 of file estimator.h.

Referenced by mappel::methods::openmp::error_bounds_profile_likelihood_stack().

9.49.3.2 IdxVecT mappel::estimator::ProfileBoundsDataStack::estimated_idxxs

List of indexes for computed parameters. Empty to compute all parameters.

Definition at line 138 of file estimator.h.

Referenced by mappel::methods::openmp::error_bounds_profile_likelihood_stack(), and mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.3 MLEDataStack mappel::estimator::ProfileBoundsDataStack::mle

Theta maximum-likelihood estimate, rllh, and Obsl stack.

Definition at line 140 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.4 IdxT mappel::estimator::ProfileBoundsDataStack::Ndata =0

size of the data stack estimated. (number of individual problem data estimates performed.)

Definition at line 144 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.5 **IdxT** mappel::estimator::ProfileBoundsDataStack::Nparams_est =0

number of parameters estimated =estimated_param_idx.n_elem.

Definition at line 143 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.6 **MatT** mappel::estimator::ProfileBoundsDataStack::profile_lb

size:[Nparams_est,Ndata] Lower bound estimated at each estimated_idx.

Definition at line 146 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.7 **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 148 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.8 **MatT** mappel::estimator::ProfileBoundsDataStack::profile_points_lb_rllh

size:[Nparams_est,Ndata] RLLH at each of the profile_points_lb

Definition at line 150 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.9 **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 149 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.10 **MatT** mappel::estimator::ProfileBoundsDataStack::profile_points_ub_rllh

size:[Nparams_est,Ndata] RLLH at each of the profile_points_ub

Definition at line 151 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.11 MatT mappel::estimator::ProfileBoundsDataStack::profile_ub

size:[Nparams_est,Ndata] Upper bound estimated at each estimated_idx.

Definition at line 147 of file estimator.h.

Referenced by mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

9.49.3.12 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 145 of file estimator.h.

Referenced by mappel::methods::openmp::error_bounds_profile_likelihood_stack(), and mappel::estimator::ThreadedEstimator< Model >::estimate_profile_bounds_stack().

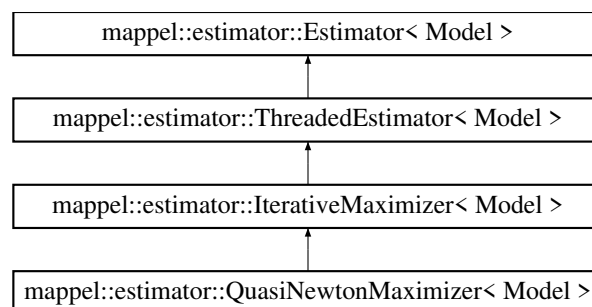
The documentation for this struct was generated from the following files:

- [estimator.h](#)
- [estimator.cpp](#)

9.50 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, 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

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, [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_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.

9.50.1 Detailed Description

```
template<class Model>
class mappel::estimator::QuasiNewtonMaximizer< Model >
```

Definition at line 601 of file estimator.h.

9.50.2 Member Typedef Documentation

9.50.2.1 `template<class Model > using mappel::estimator::QuasiNewtonMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 604 of file estimator.h.

9.50.3 Constructor & Destructor Documentation

9.50.3.1 `template<class Model > mappel::estimator::QuasiNewtonMaximizer< Model >::QuasiNewtonMaximizer (const Model & model, int max_iterations = IterativeMaximizer<Model>::DefaultIterations) [inline]`

Definition at line 606 of file estimator.h.

9.50.4 Member Function Documentation

9.50.4.1 `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()`.

9.50.4.2 `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`.

9.50.4.3 `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()`.

9.50.4.4 `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()`.

9.50.4.5 `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`.

9.50.4.6 `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`.

9.50.4.7 `template<class Model> double mappel::estimator::IterativeMaximizer< Model >::compute_profile_estimate (const ModelDataT< Model > & data, const ParamT< Model > & theta_init, const IdxVecT & fixed_idxes, StencilIT< 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\(\)](#).

9.50.4.8 `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\(\)](#).

9.50.4.9 `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\(\)](#).

9.50.4.10 `template<class Model> void mappel::estimator::Estimator< Model >::estimate_max (const ModelDataT< Model > & data, const ParamT< Model > & theta_init, MLEData & mle_data, StencilIT< 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>	MLEData 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().

9.50.4.11 `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>	MLEData 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().

9.50.4.12 `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>	MLEData 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`.

9.50.4.13 `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	<code>data</code>	Model data to estimate for
in	<code>theta_init</code>	Initial theta value.
out	<code>mle_data</code>	<code>MLEDebugData</code> recording the maximum likelihood estimate and relevant data.
out	<code>stencil</code>	[Optional] StencilT at the MLE value.

9.50.4.14 `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	<code>data</code>	Model data to estimate for
in	<code>theta_init</code>	[Optional] Initial theta value.
out	<code>mle_data</code>	<code>MLEData</code> recording the maximum likelihood estimate and relevant data.
out	<code>stencil</code>	[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.

9.50.4.15 `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>	MLEData 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.

9.50.4.16 `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.

9.50.4.17 `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().

9.50.4.18 `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().

9.50.4.19 `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().

9.50.4.20 `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_idxxs`, `mappel::estimator::ProfileBoundsDataStack::estimated_idxxs`, `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::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()`.

9.50.4.21 `template<class Model > double mappel::estimator::Estimator< Model >::estimate_profile_max (const ModelDataT< Model > & data, const IdxVecT & fixed_idxxs, 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()`.

9.50.4.22 `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_idxxs`, `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()`.

9.50.4.23 `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_idxs`, `mappel::estimator::IterativeMaximizer< Model >::get_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::last_backtrack_idxs`.

9.50.4.24 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts () const`
`[inline], [inherited]`

Run statistics.

Definition at line 276 of file `estimator.h`.

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

9.50.4.25 `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`.

9.50.4.26 `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()`.

9.50.4.27 `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_backtracks () const`
`[inline],[inherited]`

Definition at line 443 of file estimator.h.

9.50.4.28 `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_der_evals () const`
`[inline],[inherited]`

Definition at line 445 of file estimator.h.

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

9.50.4.29 `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_fun_evals () const`
`[inline],[inherited]`

Definition at line 444 of file estimator.h.

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

9.50.4.30 `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_iterations () const`
`[inline],[inherited]`

Definition at line 442 of file estimator.h.

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

9.50.4.31 `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()`.

9.50.4.32 `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()`.

9.50.4.33 `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()`.

9.50.4.34 `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()`.

9.50.4.35 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_backtracks ()`
[inherited]

9.50.4.36 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_der_evals ()`
[inherited]

9.50.4.37 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_fun_evals ()`
[inherited]

9.50.4.38 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_iterations ()`
[inherited]

9.50.4.39 `template<class Model > std::string mappel::estimator::QuasiNewtonMaximizer< Model >::name () const`
[inline],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 609 of file estimator.h.

9.50.4.40 `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\(\)](#).

9.50.4.41 `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()`.

9.50.4.42 `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()`.

9.50.4.43 `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()`.

9.50.4.44 `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().

9.50.5 Member Data Documentation

9.50.5.1 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_max_ratio = 0.50 [static], [protected], [inherited]`

Definition at line 462 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.50.5.2 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio = 1e-3 [static], [protected], [inherited]`

Definition at line 463 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::backtrack(), and mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.50.5.3 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_ratio = 0.05 [static], [protected], [inherited]`

Definition at line 461 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.50.5.4 `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 458 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio(), and mappel::estimator::IterativeMaximizer< Model >::get_stats().

9.50.5.5 `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 459 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()`.

9.50.5.6 `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::DefaultIterations = 100` [static], [inherited]

Definition at line 430 of file estimator.h.

9.50.5.7 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 301 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()`.

9.50.5.8 `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 477 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.50.5.9 `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::max_backtracks = 8` [static], [protected], [inherited]

Definition at line 464 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()`.

9.50.5.10 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::max_iterations` [protected], [inherited]

Definition at line 468 of file estimator.h.

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

9.50.5.11 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::max_threads`
`[protected], [inherited]`

Definition at line 326 of file estimator.h.

9.50.5.12 `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 456 of file estimator.h.

9.50.5.13 `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 466 of file estimator.h.

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

9.50.5.14 `template<class Model > const Model& mappel::estimator::Estimator< Model >::model [protected],`
`[inherited]`

Definition at line 296 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()`.

9.50.5.15 `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx [protected],`
`[inherited]`

Definition at line 328 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()`.

9.50.5.16 `template<class Model> int mappel::estimator::Estimator< Model >::num_estimations = 0` [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::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()`.

9.50.5.17 `template<class Model> int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 327 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()`.

9.50.5.18 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::total_backtracks = 0` [protected], [inherited]

Definition at line 472 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()`.

9.50.5.19 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::total_der_evals = 0` [protected], [inherited]

Definition at line 474 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()`.

9.50.5.20 `template<class Model> int mappel::estimator::IterativeMaximizer< Model >::total_fun_evals = 0` [protected], [inherited]

Definition at line 473 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()`.

9.50.5.21 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_iterations = 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()`.

9.50.5.22 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected], [inherited]`

Definition at line 300 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()`.

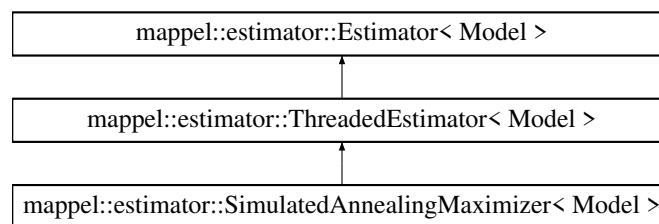
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

9.51 mappel::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](#)

9.51.1 Detailed Description

```
template<class Model>
class mappel::estimator::SimulatedAnnealingMaximizer< Model >
```

Definition at line 386 of file estimator.h.

9.51.2 Constructor & Destructor Documentation

9.51.2.1 `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 394 of file estimator.h.

9.51.3 Member Function Documentation

9.51.3.1 `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\(\)](#).

9.51.3.2 `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()`.

9.51.3.3 `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()`.

9.51.3.4 `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>	MLEData 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().

9.51.3.5 `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>	MLEData 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().

9.51.3.6 `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>	MLEData 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.

9.51.3.7 `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>	MLEDebugData recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.51.3.8 `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>	MLEData 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`.

9.51.3.9 `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>	MLEData 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`.

9.51.3.10 `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`.

9.51.3.11 `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()`.

9.51.3.12 `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()`.

9.51.3.13 `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()`.

9.51.3.14 `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_idxxs`, `mappel::estimator::ProfileBoundsDataStack::estimated_idxxs`, `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::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()`.

9.51.3.15 `template<class Model > double mappel::estimator::Estimator< Model >::estimate_profile_max (const ModelDataT< Model > & data, const IdxVecT & fixed_idxxs, 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()`.

9.51.3.16 `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_idxxs`, `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()`.

9.51.3.17 `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\(\)](#).

9.51.3.18 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts () const [inline],[inherited]`

Run statistics.

Definition at line 276 of file estimator.h.

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize\(\)](#).

9.51.3.19 `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](#).

9.51.3.20 `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](#).

9.51.3.21 `template<class Model > std::string mappel::estimator::SimulatedAnnealingMaximizer< Model >::name () const [inline],[virtual]`

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 403 of file estimator.h.

References [mappel::estimator::MLEData::rllh](#).

9.51.3.22 `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()`.

9.51.3.23 `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()`.

9.51.4 Member Data Documentation

9.51.4.1 `template<class Model > const double mappel::estimator::SimulatedAnnealingMaximizer< Model >::Default_T_Init = 100.` `[static]`

Default SA initial temperature.

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

9.51.4.2 `template<class Model > const double mappel::estimator::SimulatedAnnealingMaximizer< Model >::DefaultCoolingRate = 1.02` `[static]`

Default SA cooling rate.

Definition at line 392 of file `estimator.h`.

9.51.4.3 `template<class Model> const int mappel::estimator::SimulatedAnnealingMaximizer< Model >::DefaultNumIterations = 500 [static]`

Default number of SA iterations.

Definition at line 390 of file estimator.h.

9.51.4.4 `template<class Model> IdxVecT mappel::estimator::Estimator< Model >::exit_counts [protected], [inherited]`

Definition at line 301 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()`.

9.51.4.5 `template<class Model> int mappel::estimator::ThreadedEstimator< Model >::max_threads [protected], [inherited]`

Definition at line 326 of file estimator.h.

9.51.4.6 `template<class Model> const Model& mappel::estimator::Estimator< Model >::model [protected], [inherited]`

Definition at line 296 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()`.

9.51.4.7 `template<class Model> std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx [protected], [inherited]`

Definition at line 328 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()`.

9.51.4.8 `template<class Model> int mappel::estimator::Estimator< Model>::num_estimations = 0` [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::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()`.

9.51.4.9 `template<class Model> int mappel::estimator::ThreadedEstimator< Model>::num_threads` [protected], [inherited]

Definition at line 327 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()`.

9.51.4.10 `template<class Model> double mappel::estimator::Estimator< Model>::total_walltime = 0.` [protected], [inherited]

Definition at line 300 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](#)

9.52 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)

9.52.1 Detailed Description

[Stencil](#) for 2D free-sigma (astigmatic) models.

Definition at line 32 of file [Gauss2DsxxyModel.h](#).

9.52.2 Member Typedef Documentation

9.52.2.1 typedef Gauss2DsxxyModel::ParamT mappel::Gauss2DsxxyModel::Stencil::ParamT

Definition at line 35 of file [Gauss2DsxxyModel.h](#).

9.52.3 Constructor & Destructor Documentation

9.52.3.1 `mappel::Gauss2DsxyModel::Stencil::Stencil () [inline]`

Definition at line 47 of file Gauss2DsxyModel.h.

References `compute_derivatives()`.

Referenced by `mappel::Gauss2DsxyModel::make_stencil()`.

9.52.3.2 `mappel::Gauss2DsxyModel::Stencil::Stencil (const Gauss2DsxyModel & model, const ParamT & theta, bool _compute_derivatives = true)`

9.52.4 Member Function Documentation

9.52.4.1 `double mappel::Gauss2DsxyModel::Stencil::bg () const [inline]`

Definition at line 53 of file Gauss2DsxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxyModel::pixel_model_value()`.

9.52.4.2 `void mappel::Gauss2DsxyModel::Stencil::compute_derivatives ()`

Referenced by `Stencil()`.

9.52.4.3 `double mappel::Gauss2DsxyModel::Stencil::l () const [inline]`

Definition at line 52 of file Gauss2DsxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxyModel::pixel_grad()`, `mappel::Gauss2DsxyModel::pixel_grad2()`, `mappel::Gauss2DsxyModel::pixel_hess()`, and `mappel::Gauss2DsxyModel::pixel_model_value()`.

9.52.4.4 `double mappel::Gauss2DsxyModel::Stencil::sigmaX () const [inline]`

Definition at line 54 of file Gauss2DsxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxyModel::pixel_grad2()`, and `mappel::Gauss2DsxyModel::pixel_hess()`.

9.52.4.5 `double mappel::Gauss2DsxyModel::Stencil::sigmaY () const [inline]`

Definition at line 55 of file Gauss2DsxyModel.h.

References `operator<<`, and `theta`.

Referenced by `mappel::Gauss2DsxyModel::pixel_grad2()`, and `mappel::Gauss2DsxyModel::pixel_hess()`.

9.52.4.6 `double mappel::Gauss2DsxxyModel::Stencil::x () const [inline]`

Definition at line 50 of file Gauss2DsxxyModel.h.

References `theta`.

9.52.4.7 `double mappel::Gauss2DsxxyModel::Stencil::y () const [inline]`

Definition at line 51 of file Gauss2DsxxyModel.h.

References `theta`.

9.52.5 Friends And Related Function Documentation

9.52.5.1 `std::ostream& operator<< (std::ostream & out, const Gauss2DsxxyModel::Stencil & s) [friend]`

Referenced by `sigmaY()`.

9.52.6 Member Data Documentation

9.52.6.1 `bool mappel::Gauss2DsxxyModel::Stencil::derivatives_computed =false`

Definition at line 34 of file Gauss2DsxxyModel.h.

9.52.6.2 `VecT mappel::Gauss2DsxxyModel::Stencil::dx`

Definition at line 39 of file Gauss2DsxxyModel.h.

9.52.6.3 `VecT mappel::Gauss2DsxxyModel::Stencil::DX`

Definition at line 42 of file Gauss2DsxxyModel.h.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

9.52.6.4 `VecT mappel::Gauss2DsxxyModel::Stencil::DXS`

Definition at line 44 of file Gauss2DsxxyModel.h.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad()`, `mappel::Gauss2DsxxyModel::pixel_grad2()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

9.52.6.5 `VecT mappel::Gauss2DsxxyModel::Stencil::DXS2`

Definition at line 45 of file Gauss2DsxxyModel.h.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad2()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

9.52.6.6 VecT mappel::Gauss2DsxyModel::Stencil::DXSX

Definition at line 43 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_hess().

9.52.6.7 VecT mappel::Gauss2DsxyModel::Stencil::dy

Definition at line 39 of file Gauss2DsxyModel.h.

9.52.6.8 VecT mappel::Gauss2DsxyModel::Stencil::DY

Definition at line 42 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_grad(), and mappel::Gauss2DsxyModel::pixel_hess().

9.52.6.9 VecT mappel::Gauss2DsxyModel::Stencil::DYS

Definition at line 44 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_grad(), mappel::Gauss2DsxyModel::pixel_grad2(), and mappel::Gauss2DsxyModel::pixel_hess().

9.52.6.10 VecT mappel::Gauss2DsxyModel::Stencil::DYS2

Definition at line 45 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_grad2(), and mappel::Gauss2DsxyModel::pixel_hess().

9.52.6.11 VecT mappel::Gauss2DsxyModel::Stencil::DYSX

Definition at line 43 of file Gauss2DsxyModel.h.

9.52.6.12 VecT mappel::Gauss2DsxyModel::Stencil::DYSY

Definition at line 46 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_hess().

9.52.6.13 VecT mappel::Gauss2DsxyModel::Stencil::Gx

Definition at line 40 of file Gauss2DsxyModel.h.

9.52.6.14 VecT mappel::Gauss2DsxyModel::Stencil::Gy

Definition at line 40 of file Gauss2DsxyModel.h.

9.52.6.15 Gauss2DsxyModel const* mappel::Gauss2DsxyModel::Stencil::model

Definition at line 36 of file Gauss2DsxyModel.h.

9.52.6.16 ParamT mappel::Gauss2DsxyModel::Stencil::theta

Definition at line 38 of file Gauss2DsxyModel.h.

Referenced by bg(), l(), sigmaX(), sigmaY(), x(), and y().

9.52.6.17 VecT mappel::Gauss2DsxyModel::Stencil::X

Definition at line 41 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_grad(), mappel::Gauss2DsxyModel::pixel_grad2(), mappel::Gauss2DsxyModel::pixel_hess(), and mappel::Gauss2DsxyModel::pixel_model_value().

9.52.6.18 VecT mappel::Gauss2DsxyModel::Stencil::Y

Definition at line 41 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_grad(), mappel::Gauss2DsxyModel::pixel_grad2(), mappel::Gauss2DsxyModel::pixel_hess(), and mappel::Gauss2DsxyModel::pixel_model_value().

The documentation for this class was generated from the following file:

- [Gauss2DsxyModel.h](#)

9.53 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)

9.53.1 Detailed Description

`Stencil` for 1D variable-sigma models.

Definition at line 24 of file `Gauss1DsModel.h`.

9.53.2 Member Typedef Documentation

9.53.2.1 using `mappel::Gauss1DsModel::Stencil::ParamT` = `Gauss1DsModel::ParamT`

Definition at line 27 of file `Gauss1DsModel.h`.

9.53.3 Constructor & Destructor Documentation

9.53.3.1 `mappel::Gauss1DsModel::Stencil::Stencil ()` [inline]

Definition at line 38 of file `Gauss1DsModel.h`.

References `compute_derivatives()`.

Referenced by `mappel::Gauss1DsModel::make_stencil()`.

9.53.3.2 `mappel::Gauss1DsModel::Stencil::Stencil (const Gauss1DsModel & model, const ParamT & 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()`.

9.53.4 Member Function Documentation

9.53.4.1 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().

9.53.4.2 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().

9.53.4.3 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().

9.53.4.4 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().

9.53.4.5 double mappel::Gauss1DsModel::Stencil::x () const [inline]

Definition at line 41 of file Gauss1DsModel.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 Gauss1DsModel::Stencil & s)` [friend]

Definition at line 182 of file Gauss1DsModel.cpp.

Referenced by `sigma()`.

9.53.6 Member Data Documentation

9.53.6.1 `bool mappel::Gauss1DsModel::Stencil::derivatives_computed = false`

Definition at line 26 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, and `mappel::operator<<()`.

9.53.6.2 `VecT mappel::Gauss1DsModel::Stencil::dx`

Definition at line 31 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, and `Stencil()`.

9.53.6.3 `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_grad2()`, `mappel::Gauss1DsModel::pixel_hess()`, and `mappel::Gauss1DsModel::pixel_hess_update()`.

9.53.6.4 `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()`.

9.53.6.5 `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()`.

9.53.6.6 `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()`.

9.53.6.7 VecT mappel::Gauss1DsModel::Stencil::Gx

Definition at line 32 of file Gauss1DsModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

9.53.6.8 Gauss1DsModel const* mappel::Gauss1DsModel::Stencil::model

Definition at line 28 of file Gauss1DsModel.h.

Referenced by compute_derivatives(), and Stencil().

9.53.6.9 ParamT mappel::Gauss1DsModel::Stencil::theta

Definition at line 30 of file Gauss1DsModel.h.

Referenced by bg(), l(), mappel::operator<<(), sigma(), and x().

9.53.6.10 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](#)

9.54 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)

9.54.1 Detailed Description

`Stencil` for 2D fixed-sigma models.

Definition at line 26 of file `Gauss2DModel.h`.

9.54.2 Member Typedef Documentation**9.54.2.1 using `mappel::Gauss2DModel::Stencil::ParamT` = `Gauss2DModel::ParamT`**

Definition at line 29 of file `Gauss2DModel.h`.

9.54.3 Constructor & Destructor Documentation**9.54.3.1 `mappel::Gauss2DModel::Stencil::Stencil()` [`inline`]**

Definition at line 38 of file `Gauss2DModel.h`.

References `compute_derivatives()`.

Referenced by `mappel::Gauss2DModel::make_stencil()`.

9.54.3.2 `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()`.

9.54.4 Member Function Documentation

9.54.4.1 `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()`.

9.54.4.2 `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()`.

9.54.4.3 `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()`.

9.54.4.4 `double mappel::Gauss2DModel::Stencil::x () const [inline]`

Definition at line 41 of file Gauss2DModel.h.

References `theta`.

Referenced by `Stencil()`.

9.54.4.5 `double mappel::Gauss2DModel::Stencil::y () const` `[inline]`

Definition at line 42 of file Gauss2DModel.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 Gauss2DModel::Stencil & s)` `[friend]`

Definition at line 249 of file Gauss2DModel.cpp.

Referenced by bg().

9.54.6 Member Data Documentation

9.54.6.1 `bool mappel::Gauss2DModel::Stencil::derivatives_computed =false`

Definition at line 28 of file Gauss2DModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

9.54.6.2 `VecT mappel::Gauss2DModel::Stencil::dx`

Definition at line 33 of file Gauss2DModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

9.54.6.3 `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().

9.54.6.4 `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().

9.54.6.5 VecT mappel::Gauss2DModel::Stencil::dy

Definition at line 33 of file Gauss2DModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

9.54.6.6 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().

9.54.6.7 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().

9.54.6.8 VecT mappel::Gauss2DModel::Stencil::Gx

Definition at line 34 of file Gauss2DModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

9.54.6.9 VecT mappel::Gauss2DModel::Stencil::Gy

Definition at line 34 of file Gauss2DModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

9.54.6.10 Gauss2DModel const* mappel::Gauss2DModel::Stencil::model

Definition at line 30 of file Gauss2DModel.h.

Referenced by compute_derivatives(), and Stencil().

9.54.6.11 ParamT mappel::Gauss2DModel::Stencil::theta

Definition at line 32 of file Gauss2DModel.h.

Referenced by bg(), l(), mappel::operator<<(), x(), and y().

9.54.6.12 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().

9.54.6.13 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](#)

9.55 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)`

9.55.1 Detailed Description

[Stencil](#) for 2D scalar-sigma models.

Definition at line 29 of file Gauss2DsModel.h.

9.55.2 Member Typedef Documentation

9.55.2.1 `typedef Gauss2DsModel::ParamT mappel::Gauss2DsModel::Stencil::ParamT`

Definition at line 32 of file Gauss2DsModel.h.

9.55.3 Constructor & Destructor Documentation

9.55.3.1 `mappel::Gauss2DsModel::Stencil::Stencil () [inline]`

Definition at line 43 of file Gauss2DsModel.h.

References `compute_derivatives()`.

Referenced by `mappel::Gauss2DsModel::make_stencil()`.

9.55.3.2 `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()`.

9.55.4 Member Function Documentation

9.55.4.1 `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()`.

9.55.4.2 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().

9.55.4.3 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().

9.55.4.4 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().

9.55.4.5 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().

9.55.4.6 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().

9.55.4.7 double mappel::Gauss2DsModel::Stencil::x () const [inline]

Definition at line 46 of file Gauss2DsModel.h.

References theta.

Referenced by Stencil().

9.55.4.8 double mappel::Gauss2DsModel::Stencil::y () const [inline]

Definition at line 47 of file Gauss2DsModel.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 Gauss2DsModel::Stencil & s) [friend]

Definition at line 314 of file Gauss2DsModel.cpp.

Referenced by sigmaY().

9.55.6 Member Data Documentation

9.55.6.1 bool mappel::Gauss2DsModel::Stencil::derivatives_computed =false

Definition at line 31 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

9.55.6.2 VecT mappel::Gauss2DsModel::Stencil::dx

Definition at line 36 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

9.55.6.3 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().

9.55.6.4 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().

9.55.6.5 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().

9.55.6.6 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().

9.55.6.7 VecT mappel::Gauss2DsModel::Stencil::dy

Definition at line 36 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

9.55.6.8 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().

9.55.6.9 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().

9.55.6.10 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().

9.55.6.11 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().

9.55.6.12 VecT mappel::Gauss2DsModel::Stencil::Gx

Definition at line 37 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

9.55.6.13 VecT mappel::Gauss2DsModel::Stencil::Gy

Definition at line 37 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

9.55.6.14 Gauss2DsModel const* mappel::Gauss2DsModel::Stencil::model

Definition at line 33 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), and Stencil().

9.55.6.15 ParamT mappel::Gauss2DsModel::Stencil::theta

Definition at line 35 of file Gauss2DsModel.h.

Referenced by bg(), l(), mappel::operator<<(), sigma_ratio(), x(), and y().

9.55.6.16 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().

9.55.6.17 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](#)

9.56 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)

9.56.1 Detailed Description

[Stencil](#) for 1D fixed-sigma models.

Definition at line 29 of file [Gauss1DModel.h](#).

9.56.2 Member Typedef Documentation

9.56.2.1 using [mappel::Gauss1DModel::Stencil::ParamT](#) = [Gauss1DModel::ParamT](#)

Definition at line 32 of file [Gauss1DModel.h](#).

9.56.3 Constructor & Destructor Documentation

9.56.3.1 mappel::Gauss1DModel::Stencil () [inline]

Definition at line 41 of file Gauss1DModel.h.

References `compute_derivatives()`.

Referenced by `mappel::Gauss1DModel::make_stencil()`.

9.56.3.2 `mappel::Gauss1DModel::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()`.

9.56.4 Member Function Documentation

9.56.4.1 `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()`.

9.56.4.2 `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()`.

9.56.4.3 `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()`.

9.56.4.4 `double mappel::Gauss1DModel::Stencil::x () const` `[inline]`

Definition at line 44 of file Gauss1DModel.h.

References theta.

Referenced by Stencil().

9.56.5 Friends And Related Function Documentation

9.56.5.1 `std::ostream& operator<< (std::ostream & out, const Gauss1DModel::Stencil & s)` `[friend]`

Definition at line 164 of file Gauss1DModel.cpp.

Referenced by bg().

9.56.6 Member Data Documentation

9.56.6.1 `bool mappel::Gauss1DModel::Stencil::derivatives_computed = false`

Definition at line 31 of file Gauss1DModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

9.56.6.2 `VecT mappel::Gauss1DModel::Stencil::dx`

Definition at line 36 of file Gauss1DModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

9.56.6.3 `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().

9.56.6.4 `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, [_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
- [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](#)

9.57.1 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 311 of file `estimator.h`.

9.57.2 Constructor & Destructor Documentation

```
9.57.2.1 template<class Model > mappel::estimator::ThreadedEstimator< Model >::ThreadedEstimator ( const
Model & model )
```

Definition at line 370 of file `estimator_impl.h`.

9.57.3 Member Function Documentation

```
9.57.3.1 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\(\)](#).

9.57.3.2 `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()`.

9.57.3.3 `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()`.

9.57.3.4 `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()`.

9.57.3.5 `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()`.

9.57.3.6 `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()`.

9.57.3.7 `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>	MLEData 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()`.

9.57.3.8 `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>	MLEData 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()`.

9.57.3.9 `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>	MLEData 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`.

9.57.3.10 `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>	MLEDebugData recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.57.3.11 `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>	MLEData 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`.

9.57.3.12 `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>	MLEData 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`.

9.57.3.13 `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`.

9.57.3.14 `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()`.

9.57.3.15 `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()`.

9.57.3.16 `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()`.

9.57.3.17 `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().

9.57.3.18 `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().

9.57.3.19 `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().

9.57.3.20 `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\(\)](#).

9.57.3.21 `template<class Model> IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts () const`
`[inline],[inherited]`

Run statistics.

Definition at line 276 of file `estimator.h`.

Referenced by [mappel::estimator::IterativeMaximizer< Model >::local_profile_maximize\(\)](#).

9.57.3.22 `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](#).

9.57.3.23 `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\(\)](#).

9.57.3.24 `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\(\)](#).

9.57.3.25 `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\(\)](#).

9.57.3.26 `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\(\)](#).

9.57.4 Member Data Documentation

9.57.4.1 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected],[inherited]

Definition at line 301 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\(\)](#).

9.57.4.2 `template<class Model> int mappel::estimator::ThreadedEstimator< Model >::max_threads` [protected]

Definition at line 326 of file estimator.h.

9.57.4.3 `template<class Model> const Model& mappel::estimator::Estimator< Model >::model` [protected], [inherited]

Definition at line 296 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()`.

9.57.4.4 `template<class Model> std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` [protected]

Definition at line 328 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()`.

9.57.4.5 `template<class Model> int mappel::estimator::Estimator< Model >::num_estimations = 0` [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::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()`.

9.57.4.6 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads` [protected]

Definition at line 327 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()`.

9.57.4.7 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` [protected], [inherited]

Definition at line 300 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()`.

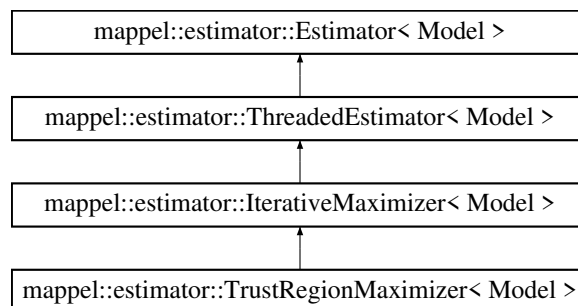
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

9.58 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, `MLEDData` &mle_data, `StencilT`< Model > &mle_stencil) override
- void `compute_estimate_debug` (const `ModelDataT`< Model > &data, const `ParamT`< Model > &theta_init, `MLEDDebugData` &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](#)s
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.

9.58.1 Detailed Description

```
template<class Model>
class mappel::estimator::TrustRegionMaximizer< Model >
```

Definition at line 616 of file estimator.h.

9.58.2 Member Typedef Documentation

9.58.2.1 `template<class Model> using mappel::estimator::TrustRegionMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 619 of file estimator.h.

9.58.3 Constructor & Destructor Documentation

9.58.3.1 `template<class Model> mappel::estimator::TrustRegionMaximizer< Model >::TrustRegionMaximizer (const Model & model, int max_iterations = IterativeMaximizer<Model>::DefaultIterations) [inline]`

Definition at line 629 of file estimator.h.

9.58.4 Member Function Documentation

9.58.4.1 `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()`.

9.58.4.2 `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`.

9.58.4.3 `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\(\)](#).

9.58.4.4 `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\(\)](#).

9.58.4.5 `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](#).

9.58.4.6 `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](#).

9.58.4.7 `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\(\)](#).

9.58.4.8 `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\(\)](#).

9.58.4.9 `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\(\)](#).

9.58.4.10 `template<class Model > void mappel::estimator::Estimator< Model >::estimate_max (const ModelIDataT< 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>	MLEData 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()`.

9.58.4.11 `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>	MLEData 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()`.

9.58.4.12 `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>	MLEData 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`.

9.58.4.13 `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>	MLEDebugData recording the maximum likelihood estimate and relevant data.
out	<i>stencil</i>	[Optional] StencilT at the MLE value.

9.58.4.14 `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>	MLEData 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`.

9.58.4.15 `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>	MLEData 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`.

9.58.4.16 `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`.

9.58.4.17 `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()`.

9.58.4.18 `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()`.

9.58.4.19 `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()`.

9.58.4.20 `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().

9.58.4.21 `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().

9.58.4.22 `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().

9.58.4.23 `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.

9.58.4.24 `template<class Model> IdxVecT mappel::estimator::Estimator< Model >::get_exit_counts () const`
`[inline], [inherited]`

Run statistics.

Definition at line 276 of file `estimator.h`.

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

9.58.4.25 `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`.

9.58.4.26 `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()`.

9.58.4.27 `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_backtracks () const`
`[inline],[inherited]`

Definition at line 443 of file estimator.h.

9.58.4.28 `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_der_evals () const`
`[inline],[inherited]`

Definition at line 445 of file estimator.h.

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

9.58.4.29 `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_fun_evals () const`
`[inline],[inherited]`

Definition at line 444 of file estimator.h.

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

9.58.4.30 `template<class Model> int mappel::estimator::IterativeMaximizer< Model>::get_total_iterations () const`
`[inline],[inherited]`

Definition at line 442 of file estimator.h.

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

9.58.4.31 `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()`.

9.58.4.32 `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()`.

9.58.4.33 `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()`.

9.58.4.34 `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()`.

9.58.4.35 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_backtracks ()`
[inherited]

9.58.4.36 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_der_evals ()`
[inherited]

9.58.4.37 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_fun_evals ()`
[inherited]

9.58.4.38 `template<class Model > double mappel::estimator::IterativeMaximizer< Model >::mean_iterations ()`
[inherited]

9.58.4.39 `template<class Model> std::string mappel::estimator::TrustRegionMaximizer< Model >::name () const`
[inline],[virtual]

Implements [mappel::estimator::Estimator< Model >](#).

Definition at line 632 of file estimator.h.

9.58.4.40 `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\(\)](#).

9.58.4.41 `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()`.

9.58.4.42 `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()`.

9.58.4.43 `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()`.

9.58.4.44 `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().

9.58.5 Member Data Documentation

9.58.5.1 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_max_ratio = 0.50 [static], [protected], [inherited]`

Definition at line 462 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.58.5.2 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_linear_step_ratio = 1e-3 [static], [protected], [inherited]`

Definition at line 463 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::backtrack(), and mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.58.5.3 `template<class Model > const double mappel::estimator::IterativeMaximizer< Model >::backtrack_min_ratio = 0.05 [static], [protected], [inherited]`

Definition at line 461 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::profile_bound_backtrack().

9.58.5.4 `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 458 of file estimator.h.

Referenced by mappel::estimator::IterativeMaximizer< Model >::convergence_test_grad_ratio(), and mappel::estimator::IterativeMaximizer< Model >::get_stats().

9.58.5.5 `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 459 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()`.

9.58.5.6 `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 627 of file estimator.h.

9.58.5.7 `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::DefaultIterations = 100` [static], [inherited]

Definition at line 430 of file estimator.h.

9.58.5.8 `template<class Model > IdxVecT mappel::estimator::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 301 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()`.

9.58.5.9 `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 477 of file estimator.h.

Referenced by `mappel::estimator::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::estimator::IterativeMaximizer< Model >::record_run_statistics()`.

9.58.5.10 `template<class Model > const int mappel::estimator::IterativeMaximizer< Model >::max_backtracks = 8` [static], [protected], [inherited]

Definition at line 464 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()`.

9.58.5.11 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::max_iterations`
`[protected], [inherited]`

Definition at line 468 of file estimator.h.

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

9.58.5.12 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::max_threads`
`[protected], [inherited]`

Definition at line 326 of file estimator.h.

9.58.5.13 `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 456 of file estimator.h.

9.58.5.14 `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 466 of file estimator.h.

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

9.58.5.15 `template<class Model > const Model& mappel::estimator::Estimator< Model >::model [protected],`
`[inherited]`

Definition at line 296 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()`.

9.58.5.16 `template<class Model > std::mutex mappel::estimator::ThreadedEstimator< Model >::mtx` [protected],
[inherited]

Definition at line 328 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()`.

9.58.5.17 `template<class Model > int mappel::estimator::Estimator< Model >::num_estimations = 0` [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::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()`.

9.58.5.18 `template<class Model > int mappel::estimator::ThreadedEstimator< Model >::num_threads`
[protected], [inherited]

Definition at line 327 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()`.

9.58.5.19 `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 621 of file estimator.h.

9.58.5.20 `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 622 of file estimator.h.

9.58.5.21 `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 623 of file estimator.h.

9.58.5.22 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_backtracks = 0`
`[protected], [inherited]`

Definition at line 472 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()`.

9.58.5.23 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_der_evals = 0`
`[protected], [inherited]`

Definition at line 474 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()`.

9.58.5.24 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_fun_evals = 0`
`[protected], [inherited]`

Definition at line 473 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()`.

9.58.5.25 `template<class Model > int mappel::estimator::IterativeMaximizer< Model >::total_iterations = 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()`.

9.58.5.26 `template<class Model > double mappel::estimator::Estimator< Model >::total_walltime = 0.` `[protected], [inherited]`

Definition at line 300 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()`.

9.58.5.27 `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 625 of file estimator.h.

9.58.5.28 `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 624 of file estimator.h.

9.58.5.29 `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 626 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)

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)

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)

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

MatT	<code>obsI</code>	Observed Fisher information matrix at theta.
<code>double</code>	<code>rllh</code>	RLLH at theta.
VecT	<code>theta</code>	Theta estimate.

10.4.2.2 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 65 of file estimator.h.

Class Members

IdxT	Ndata	Number of data estimates.
CubeT	obsI	Observed Fisher information matrix stack. size:[Nparams,Nparams,Ndata].
VecT	rllh	RLLH stack. size:[Ndata].
MatT	theta	Theta estimate stack. size:[Nparams,Ndata].

10.4.2.3 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 76 of file estimator.h.

Class Members

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

10.4.2.4 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 (ouput) parameters to give output parameters context.

Definition at line 115 of file estimator.h.

Class Members

IdxT	estimated_idx	Index of single parameter to estimate for.
MLEData	mle	Theta maximum-likelihood estimate, rllh, and ObsI.
IdxT	Nseq_lb	Number of points in sequence_lb.
IdxT	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.
MatT	sequence_lb	size:[NumParams,Nseq_lb] Sequence of evaluated points for lb estimate (including theta mle as initial point)

Class Members

VecT	sequence_lb_rllh	size:[Nseq_lb] RLLH at each of the sequence_lb points
MatT	sequence_ub	size:[NumParams,Nseq_ub] Sequence of evaluated points for ub estimate (including theta mle as initial point)
VecT	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 ($-\chi^2_{\text{inv}}(\text{confidence},1)/2$)

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::Gauss2DMAP](#)
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 Gauss2DMap.

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/MCMCAdaptor2D.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/MCMCAdaptor2Ds.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,bg,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< Gauss2DsxModel, 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< Gauss2DsxModel, 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< Gauss2DsxModel, 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)`

10.33.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsxModel.

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)

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.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

2018-2019

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 OMPExcceptionCatcher.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_::OMPExcceptionCatcher< _dummy >](#)

Namespaces

- [omp_exception_catcher](#)
- [omp_exception_catcher::impl_](#)

Typedefs

- using [omp_exception_catcher::OMPExcceptionCatcher](#) = [impl_::OMPExcceptionCatcher<>](#)

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 DOT 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::OMPEXceptionCatcher` 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/OMPEXceptionCatcher/OMPEXceptionCatcher.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 > &crlb_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, estimator::MLEDataStack &mle_data_stack, StatsT &stats)
- `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 &obsI_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 Model↵
DataStackT< Model > &image, estimator::ProfileBoundsData &est, StatsT &stats)`
- `template<class Model >`
`void mappel::methods::openmp::error_bounds_profile_likelihood_parallel (const Model &model, const Model↵
DataT< Model > &image, estimator::ProfileBoundsData &est)`
- `template<class Model >`
`void mappel::methods::openmp::error_bounds_profile_likelihood_stack (const Model &model, const ModelData↵
StackT< Model > &image, estimator::ProfileBoundsDataStack &est, StatsT &stats)`
- `template<class Model >`
`void mappel::methods::openmp::error_bounds_profile_likelihood_stack (const Model &model, const ModelData↵
StackT< 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 Image↵
StackT< 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 palatalization 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

2014-2017

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<PoissonGaussianNoise2D↵
Objective,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

<code>out</code>	<code>image</code>	An image to populate.
<code>in</code>	<code>theta</code>	The parameter values to us
<code>in, out</code>	<code>rng</code>	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.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 using 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)

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=1↵ E-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 = backtrack_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, [118](#)
- Abort
 - omp_exception_catcher, [90](#)
- aposteriori_objective
 - mappel::methods, [58](#)
- ArrayShapeError
 - mappel::ArrayShapeError, [91](#)
- ArraySizeError
 - mappel::ArraySizeError, [91](#)
- backtrack
 - mappel::estimator::IterativeMaximizer, [572](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [725](#)
 - mappel::estimator::NewtonMaximizer, [746](#)
 - mappel::estimator::QuasiNewtonMaximizer, [811](#)
 - mappel::estimator::TrustRegionMaximizer, [880](#)
- backtrack_idxxs
 - mappel::estimator::IterativeMaximizer::Maximizer↔Data, [612](#)
- backtrack_max_ratio
 - mappel::estimator::IterativeMaximizer, [585](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [738](#)
 - mappel::estimator::NewtonMaximizer, [759](#)
 - mappel::estimator::QuasiNewtonMaximizer, [824](#)
 - mappel::estimator::TrustRegionMaximizer, [894](#)
- backtrack_min_linear_step_ratio
 - mappel::estimator::IterativeMaximizer, [585](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [738](#)
 - mappel::estimator::NewtonMaximizer, [759](#)
 - mappel::estimator::QuasiNewtonMaximizer, [824](#)
 - mappel::estimator::TrustRegionMaximizer, [894](#)
- backtrack_min_ratio
 - mappel::estimator::IterativeMaximizer, [586](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [738](#)
 - mappel::estimator::NewtonMaximizer, [759](#)
 - mappel::estimator::QuasiNewtonMaximizer, [824](#)
 - mappel::estimator::TrustRegionMaximizer, [894](#)
- beta2_prior_grad
 - mappel, [29](#)
- beta2_prior_grad2
 - mappel, [29](#)
- beta_prior_grad
 - mappel, [29](#)
- beta_prior_grad2
 - mappel, [29](#)
- bg
 - mappel::Gauss1DModel::Stencil, [861](#)
 - mappel::Gauss1DsModel::Stencil, [847](#)
 - mappel::Gauss2DModel::Stencil, [851](#)
 - mappel::Gauss2DsModel::Stencil, [855](#)
 - mappel::Gauss2DsxyModel::Stencil, [842](#)
- BoolVecT
 - mappel, [26](#)
- BoolT
 - mappel, [26](#)
- bound_step
 - mappel::estimator::subroutine, [51](#)
- bound_theta
 - mappel::Gauss1DModel, [134](#)
 - mappel::Gauss1DMLE, [163](#)
 - mappel::Gauss1DModel, [192](#)
 - mappel::Gauss1DsMAP, [220](#)
 - mappel::Gauss1DsMLE, [249](#)
 - mappel::Gauss1DsModel, [278](#)
 - mappel::Gauss2DModel, [308](#)
 - mappel::Gauss2DMLE, [339](#)
 - mappel::Gauss2DModel, [370](#)
 - mappel::Gauss2DsMAP, [401](#)
 - mappel::Gauss2DsMLE, [435](#)
 - mappel::Gauss2DsModel, [468](#)
 - mappel::Gauss2DsxyMAP, [498](#)
 - mappel::Gauss2DsxyModel, [526](#)
 - mappel::MAPEstimator, [594](#)
 - mappel::MCMCAdaptor1Ds, [638](#)
 - mappel::MCMCAdaptor1D, [619](#)
 - mappel::MCMCAdaptor2Ds, [677](#)
 - mappel::MCMCAdaptor2D, [658](#)
 - mappel::MLEstimator, [708](#)
 - mappel::PointEmitterModel, [772](#)
- bounded_theta
 - mappel::Gauss1DModel, [134](#)
 - mappel::Gauss1DMLE, [163](#)
 - mappel::Gauss1DModel, [192](#)
 - mappel::Gauss1DsMAP, [220](#)
 - mappel::Gauss1DsMLE, [249](#)
 - mappel::Gauss1DsModel, [278](#)
 - mappel::Gauss2DModel, [308](#)
 - mappel::Gauss2DMLE, [339](#)
 - mappel::Gauss2DModel, [370](#)
 - mappel::Gauss2DsMAP, [401](#)
 - mappel::Gauss2DsMLE, [435](#)
 - mappel::Gauss2DsModel, [468](#)
 - mappel::Gauss2DsxyMAP, [498](#)
 - mappel::Gauss2DsxyModel, [526](#)
 - mappel::MAPEstimator, [594](#)
 - mappel::MCMCAdaptor1Ds, [638](#)
 - mappel::MCMCAdaptor1D, [619](#)
 - mappel::MCMCAdaptor2Ds, [677](#)
 - mappel::MCMCAdaptor2D, [658](#)
 - mappel::MLEstimator, [708](#)
 - mappel::PointEmitterModel, [772](#)

bounded_theta_stack

- mappel::Gauss1DMap, 135
- mappel::Gauss1DMLE, 164
- mappel::Gauss1DModel, 192
- mappel::Gauss1DsMap, 220
- mappel::Gauss1DsMLE, 250
- mappel::Gauss1DsModel, 279
- mappel::Gauss2DMap, 308
- mappel::Gauss2DMLE, 340
- mappel::Gauss2DModel, 370
- mappel::Gauss2DsMap, 401
- mappel::Gauss2DsMLE, 435
- mappel::Gauss2DsModel, 468
- mappel::Gauss2DsxyMap, 499
- mappel::Gauss2DsxyModel, 526
- mappel::MAPEstimator, 594
- mappel::MCMCAdaptor1Ds, 638
- mappel::MCMCAdaptor1D, 619
- mappel::MCMCAdaptor2Ds, 678
- mappel::MCMCAdaptor2D, 658
- mappel::MLEstimator, 708
- mappel::PointEmitterModel, 772

bounds_epsilon

- mappel::Gauss1DMap, 149
- mappel::Gauss1DMLE, 178
- mappel::Gauss1DModel, 206
- mappel::Gauss1DsMap, 236
- mappel::Gauss1DsMLE, 265
- mappel::Gauss1DsModel, 294
- mappel::Gauss2DMap, 324
- mappel::Gauss2DMLE, 355
- mappel::Gauss2DModel, 386
- mappel::Gauss2DsMap, 419
- mappel::Gauss2DsMLE, 452
- mappel::Gauss2DsModel, 484
- mappel::Gauss2DsxyMap, 513
- mappel::Gauss2DsxyModel, 538
- mappel::MAPEstimator, 603
- mappel::MCMCAdaptor1Ds, 648
- mappel::MCMCAdaptor1D, 629
- mappel::MCMCAdaptor2Ds, 687
- mappel::MCMCAdaptor2D, 668
- mappel::MLEstimator, 717
- mappel::PointEmitterModel, 780

CGaussHeuristicEstimator

- mappel::estimator::CGaussHeuristicEstimator, 93

CGaussMLE

- mappel::estimator::CGaussMLE, 105

candidate

- mappel::mcmc::MCMCDebugData, 701

candidate_rllh

- mappel::mcmc::MCMCDebugData, 701

cgauss_compute_estimate

- mappel, 29, 30

cgauss_compute_estimate_debug

- mappel, 30

cgauss_heuristic_compute_estimate

- mappel, 30, 31

check_image_shape

- mappel::Gauss1DMap, 135
- mappel::Gauss1DMLE, 164
- mappel::Gauss1DModel, 192
- mappel::Gauss1DsMap, 220, 221
- mappel::Gauss1DsMLE, 250
- mappel::Gauss1DsModel, 279
- mappel::Gauss2DMap, 309
- mappel::Gauss2DMLE, 340
- mappel::Gauss2DModel, 371
- mappel::Gauss2DsMap, 402
- mappel::Gauss2DsMLE, 435
- mappel::Gauss2DsModel, 468
- mappel::Gauss2DsxyMap, 499
- mappel::Gauss2DsxyModel, 526, 527
- mappel::ImageFormat1DBase, 560
- mappel::ImageFormat2DBase, 566
- mappel::PoissonNoise1DObjective, 790
- mappel::PoissonNoise2DObjective, 797

check_lower_bound_hyperparameter

- mappel, 31

check_param_shape

- mappel::Gauss1DMap, 135
- mappel::Gauss1DMLE, 164
- mappel::Gauss1DModel, 192, 193
- mappel::Gauss1DsMap, 221
- mappel::Gauss1DsMLE, 250
- mappel::Gauss1DsModel, 279
- mappel::Gauss2DMap, 309
- mappel::Gauss2DMLE, 340
- mappel::Gauss2DModel, 371
- mappel::Gauss2DsMap, 402
- mappel::Gauss2DsMLE, 435, 436
- mappel::Gauss2DsModel, 468
- mappel::Gauss2DsxyMap, 499
- mappel::Gauss2DsxyModel, 527
- mappel::MAPEstimator, 594
- mappel::MCMCAdaptor1Ds, 639
- mappel::MCMCAdaptor1D, 620
- mappel::MCMCAdaptor2Ds, 678
- mappel::MCMCAdaptor2D, 658
- mappel::MLEstimator, 708
- mappel::PointEmitterModel, 772

check_positive_hyperparameter

- mappel, 31

check_psf_sigma

- mappel::Gauss1DMap, 135, 136
- mappel::Gauss1DMLE, 164, 165
- mappel::Gauss1DModel, 193

- mappel::Gauss1DsMAP, 221
- mappel::Gauss1DsMLE, 250, 251
- mappel::Gauss1DsModel, 279, 280
- mappel::Gauss2DMAP, 309
- mappel::Gauss2DMLE, 340, 341
- mappel::Gauss2DModel, 371
- mappel::Gauss2DsMAP, 402
- mappel::Gauss2DsMLE, 436
- mappel::Gauss2DsModel, 469
- mappel::Gauss2DsxyMAP, 499, 500
- mappel::Gauss2DsxyModel, 527
- mappel::MAPEstimator, 595
- mappel::MCMCAdaptor1Ds, 639
- mappel::MCMCAdaptor1D, 620
- mappel::MCMCAdaptor2Ds, 678
- mappel::MCMCAdaptor2D, 658, 659
- mappel::MLEstimator, 708, 709
- mappel::PointEmitterModel, 772, 773
- check_size
 - mappel::Gauss1DMAP, 136
 - mappel::Gauss1DMLE, 165
 - mappel::Gauss1DModel, 193
 - mappel::Gauss1DsMAP, 221
 - mappel::Gauss1DsMLE, 251
 - mappel::Gauss1DsModel, 280
 - mappel::Gauss2DMAP, 310
 - mappel::Gauss2DMLE, 341
 - mappel::Gauss2DModel, 372
 - mappel::Gauss2DsMAP, 403
 - mappel::Gauss2DsMLE, 436
 - mappel::Gauss2DsModel, 469
 - mappel::Gauss2DsxyMAP, 500
 - mappel::Gauss2DsxyModel, 527
 - mappel::ImageFormat1DBase, 560
 - mappel::ImageFormat2DBase, 566
 - mappel::PoissonNoise1DObjective, 790
 - mappel::PoissonNoise2DObjective, 797
- 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, 93
 - mappel::estimator::CGaussMLE, 106
 - mappel::estimator::Estimator, 118
 - mappel::estimator::HeuristicEstimator, 546
 - mappel::estimator::IterativeMaximizer, 572
 - mappel::estimator::NewtonDiagonalMaximizer, 725
 - mappel::estimator::NewtonMaximizer, 746
 - mappel::estimator::QuasiNewtonMaximizer, 811
 - mappel::estimator::SimulatedAnnealingMaximizer, 830
 - mappel::estimator::ThreadedEstimator, 865
 - mappel::estimator::TrustRegionMaximizer, 880
- ClockT
 - estimator.h, 905
- compute_D_scale
 - mappel::estimator::subroutine, 52
- compute_bound_scaling_vec
 - mappel::estimator::subroutine, 51
- compute_cauchy_point
 - mappel::estimator::subroutine, 52
- compute_derivatives
 - mappel::Gauss1DModel::Stencil, 861
 - mappel::Gauss1DsModel::Stencil, 847
 - mappel::Gauss2DModel::Stencil, 851
 - mappel::Gauss2DsModel::Stencil, 855
 - mappel::Gauss2DsxyModel::Stencil, 842
- compute_estimate
 - mappel::estimator::Estimator, 118
 - mappel::estimator::IterativeMaximizer, 573
 - mappel::estimator::NewtonDiagonalMaximizer, 725
 - mappel::estimator::NewtonMaximizer, 746
 - mappel::estimator::QuasiNewtonMaximizer, 811
 - mappel::estimator::ThreadedEstimator, 865
 - mappel::estimator::TrustRegionMaximizer, 880
- compute_estimate_debug
 - mappel::estimator::CGaussHeuristicEstimator, 93
 - mappel::estimator::Estimator, 119
 - mappel::estimator::HeuristicEstimator, 546
 - mappel::estimator::IterativeMaximizer, 573
 - mappel::estimator::NewtonDiagonalMaximizer, 725
 - mappel::estimator::NewtonMaximizer, 746
 - mappel::estimator::QuasiNewtonMaximizer, 811
 - mappel::estimator::ThreadedEstimator, 866
 - mappel::estimator::TrustRegionMaximizer, 881
- compute_initial_trust_radius
 - mappel::estimator::subroutine, 52
- compute_max_sigma_ratio
 - mappel::Gauss2DsMAP, 403
 - mappel::Gauss2DsMLE, 436
 - mappel::Gauss2DsModel, 469
 - mappel::Gauss2DsxyModel, 528
- compute_posterior_credible

- mappel::mcmc, 54
- compute_profile_bound
 - mappel::estimator::CGaussHeuristicEstimator, 94
 - mappel::estimator::CGaussMLE, 106
 - mappel::estimator::Estimator, 119
 - mappel::estimator::HeuristicEstimator, 546
 - mappel::estimator::IterativeMaximizer, 573
 - mappel::estimator::NewtonDiagonalMaximizer, 726
 - mappel::estimator::NewtonMaximizer, 747
 - mappel::estimator::QuasiNewtonMaximizer, 812
 - mappel::estimator::SimulatedAnnealingMaximizer, 830
 - mappel::estimator::ThreadedEstimator, 866
 - mappel::estimator::TrustRegionMaximizer, 881
- compute_profile_bound_debug
 - mappel::estimator::CGaussHeuristicEstimator, 94
 - mappel::estimator::CGaussMLE, 106
 - mappel::estimator::Estimator, 119
 - mappel::estimator::HeuristicEstimator, 547
 - mappel::estimator::IterativeMaximizer, 574
 - mappel::estimator::NewtonDiagonalMaximizer, 726
 - mappel::estimator::NewtonMaximizer, 747
 - mappel::estimator::QuasiNewtonMaximizer, 812
 - mappel::estimator::SimulatedAnnealingMaximizer, 831
 - mappel::estimator::ThreadedEstimator, 866
 - mappel::estimator::TrustRegionMaximizer, 881
- compute_profile_estimate
 - mappel::estimator::CGaussHeuristicEstimator, 94
 - mappel::estimator::CGaussMLE, 106
 - mappel::estimator::Estimator, 119
 - mappel::estimator::HeuristicEstimator, 547
 - mappel::estimator::IterativeMaximizer, 574
 - mappel::estimator::NewtonDiagonalMaximizer, 726
 - mappel::estimator::NewtonMaximizer, 747
 - mappel::estimator::QuasiNewtonMaximizer, 812
 - mappel::estimator::ThreadedEstimator, 866
 - mappel::estimator::TrustRegionMaximizer, 882
- compute_quadratic_model_value
 - mappel::estimator::subroutine, 52
- compute_scaled_problem
 - mappel::estimator::subroutine, 52
- confidence
 - mappel::estimator::ProfileBoundsData, 802
 - mappel::estimator::ProfileBoundsDataStack, 805
 - mappel::mcmc::MCMCData, 696
 - mappel::mcmc::MCMCDataStack, 699
- Continue
 - omp_exception_catcher, 90
- convergence_min_function_change_ratio
 - mappel::estimator::IterativeMaximizer, 586
 - mappel::estimator::NewtonDiagonalMaximizer, 738
 - mappel::estimator::NewtonMaximizer, 759
 - mappel::estimator::QuasiNewtonMaximizer, 824
- mappel::estimator::TrustRegionMaximizer, 894
- convergence_min_step_size_ratio
 - mappel::estimator::IterativeMaximizer, 586
 - mappel::estimator::NewtonDiagonalMaximizer, 738
 - mappel::estimator::NewtonMaximizer, 759
 - mappel::estimator::QuasiNewtonMaximizer, 824
 - mappel::estimator::TrustRegionMaximizer, 894
- convergence_min_trust_radius
 - mappel::estimator::TrustRegionMaximizer, 895
- convergence_test_grad_ratio
 - mappel::estimator::IterativeMaximizer, 574
 - mappel::estimator::NewtonDiagonalMaximizer, 727
 - mappel::estimator::NewtonMaximizer, 748
 - mappel::estimator::QuasiNewtonMaximizer, 813
 - mappel::estimator::TrustRegionMaximizer, 882
- convergence_test_step_size
 - mappel::estimator::IterativeMaximizer, 575
 - mappel::estimator::NewtonDiagonalMaximizer, 727
 - mappel::estimator::NewtonMaximizer, 748
 - mappel::estimator::QuasiNewtonMaximizer, 813
 - mappel::estimator::TrustRegionMaximizer, 882
- CoordIdxT
 - PoissonGaussianNoise2DObjective, 785
- CoordStackT
 - PoissonGaussianNoise2DObjective, 785
- CoordT
 - PoissonGaussianNoise2DObjective, 785
- 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, 696
 - mappel::mcmc::MCMCDataStack, 699
- credible_ub
 - mappel::mcmc::MCMCData, 697
 - mappel::mcmc::MCMCDataStack, 699
- CubeT
 - mappel, 27
- current_stencil
 - mappel::estimator::IterativeMaximizer::Maximizer←Data, 612
- DXS2
 - mappel::Gauss1DsModel::Stencil, 848
 - mappel::Gauss2DsModel::Stencil, 857
 - mappel::Gauss2DsxyModel::Stencil, 843
- DXSX

- mappel::Gauss1DsModel::Stencil, [848](#)
- mappel::Gauss2DsModel::Stencil, [858](#)
- mappel::Gauss2DsxyModel::Stencil, [843](#)
- DXS
 - mappel::Gauss1DModel::Stencil, [862](#)
 - mappel::Gauss1DsModel::Stencil, [848](#)
 - mappel::Gauss2DModel::Stencil, [852](#)
 - mappel::Gauss2DsModel::Stencil, [857](#)
 - mappel::Gauss2DsxyModel::Stencil, [843](#)
- DYS2
 - mappel::Gauss2DsModel::Stencil, [858](#)
 - mappel::Gauss2DsxyModel::Stencil, [844](#)
- DYSX
 - mappel::Gauss2DsxyModel::Stencil, [844](#)
- DYSY
 - mappel::Gauss2DsModel::Stencil, [858](#)
 - mappel::Gauss2DsxyModel::Stencil, [844](#)
- DYS
 - mappel::Gauss2DModel::Stencil, [853](#)
 - mappel::Gauss2DsModel::Stencil, [858](#)
 - mappel::Gauss2DsxyModel::Stencil, [844](#)
- debug_internal_sum_model_x
 - mappel::Gauss2DMAP, [310](#)
 - mappel::Gauss2DMLE, [341](#)
 - mappel::Gauss2DModel, [372](#)
 - mappel::Gauss2DsMAP, [403](#)
 - mappel::Gauss2DsMLE, [437](#)
 - mappel::Gauss2DsModel, [469](#)
- debug_internal_sum_model_y
 - mappel::Gauss2DMAP, [310](#)
 - mappel::Gauss2DMLE, [341](#)
 - mappel::Gauss2DModel, [372](#)
 - mappel::Gauss2DsMAP, [403](#)
 - mappel::Gauss2DsMLE, [437](#)
 - mappel::Gauss2DsModel, [469](#)
- Default_T_Init
 - mappel::estimator::SimulatedAnnealingMaximizer, [838](#)
- DefaultConfidenceLevel
 - mappel::Gauss1DMAP, [149](#)
 - mappel::Gauss1DMLE, [178](#)
 - mappel::Gauss1DModel, [206](#)
 - mappel::Gauss1DsMAP, [236](#)
 - mappel::Gauss1DsMLE, [265](#)
 - mappel::Gauss1DsModel, [294](#)
 - mappel::Gauss2DMAP, [324](#)
 - mappel::Gauss2DMLE, [355](#)
 - mappel::Gauss2DModel, [386](#)
 - mappel::Gauss2DsMAP, [419](#)
 - mappel::Gauss2DsMLE, [452](#)
 - mappel::Gauss2DsModel, [484](#)
 - mappel::Gauss2DsxyMAP, [513](#)
 - mappel::Gauss2DsxyModel, [538](#)
 - mappel::MAPEstimator, [603](#)
- mappel::MCMCAdaptor1Ds, [648](#)
- mappel::MCMCAdaptor1D, [629](#)
- mappel::MCMCAdaptor2Ds, [687](#)
- mappel::MCMCAdaptor2D, [668](#)
- mappel::MLEstimator, [717](#)
- mappel::PointEmitterModel, [780](#)
- DefaultCoolingRate
 - mappel::estimator::SimulatedAnnealingMaximizer, [838](#)
- DefaultEstimatorMethod
 - mappel::Gauss1DMAP, [150](#)
 - mappel::Gauss1DMLE, [179](#)
 - mappel::Gauss1DModel, [207](#)
 - mappel::Gauss1DsMAP, [236](#)
 - mappel::Gauss1DsMLE, [265](#)
 - mappel::Gauss1DsModel, [294](#)
 - mappel::Gauss2DMAP, [324](#)
 - mappel::Gauss2DMLE, [355](#)
 - mappel::Gauss2DModel, [386](#)
 - mappel::Gauss2DsMAP, [419](#)
 - mappel::Gauss2DsMLE, [452](#)
 - mappel::Gauss2DsModel, [484](#)
 - mappel::Gauss2DsxyMAP, [513](#)
 - mappel::Gauss2DsxyModel, [539](#)
 - mappel::MAPEstimator, [603](#)
 - mappel::MCMCAdaptor1Ds, [649](#)
 - mappel::MCMCAdaptor1D, [629](#)
 - mappel::MCMCAdaptor2Ds, [688](#)
 - mappel::MCMCAdaptor2D, [668](#)
 - mappel::MLEstimator, [717](#)
 - mappel::PointEmitterModel, [780](#)
- DefaultIterations
 - mappel::estimator::CGaussMLE, [114](#)
 - mappel::estimator::IterativeMaximizer, [586](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [739](#)
 - mappel::estimator::NewtonMaximizer, [760](#)
 - mappel::estimator::QuasiNewtonMaximizer, [825](#)
 - mappel::estimator::TrustRegionMaximizer, [895](#)
- DefaultMCMCBurnin
 - mappel::Gauss1DMAP, [150](#)
 - mappel::Gauss1DMLE, [179](#)
 - mappel::Gauss1DModel, [207](#)
 - mappel::Gauss1DsMAP, [236](#)
 - mappel::Gauss1DsMLE, [265](#)
 - mappel::Gauss1DsModel, [294](#)
 - mappel::Gauss2DMAP, [325](#)
 - mappel::Gauss2DMLE, [356](#)
 - mappel::Gauss2DModel, [386](#)
 - mappel::Gauss2DsMAP, [419](#)
 - mappel::Gauss2DsMLE, [452](#)
 - mappel::Gauss2DsModel, [484](#)
 - mappel::Gauss2DsxyMAP, [513](#)
 - mappel::Gauss2DsxyModel, [539](#)
 - mappel::MAPEstimator, [603](#)

- mappel::MCMCAdaptor1Ds, 649
- mappel::MCMCAdaptor1D, 630
- mappel::MCMCAdaptor2Ds, 688
- mappel::MCMCAdaptor2D, 668
- mappel::MLEstimator, 717
- mappel::PointEmitterModel, 780
- DefaultMCMCNumSamples
 - mappel::Gauss1DMAP, 150
 - mappel::Gauss1DMLE, 179
 - mappel::Gauss1DModel, 207
 - mappel::Gauss1DsMAP, 236
 - mappel::Gauss1DsMLE, 265
 - mappel::Gauss1DsModel, 294
 - mappel::Gauss2DMAP, 325
 - mappel::Gauss2DMLE, 356
 - mappel::Gauss2DModel, 386
 - mappel::Gauss2DsMAP, 419
 - mappel::Gauss2DsMLE, 452
 - mappel::Gauss2DsModel, 485
 - mappel::Gauss2DsxyMAP, 513
 - mappel::Gauss2DsxyModel, 539
 - mappel::MAPEstimator, 603
 - mappel::MCMCAdaptor1Ds, 649
 - mappel::MCMCAdaptor1D, 630
 - mappel::MCMCAdaptor2Ds, 688
 - mappel::MCMCAdaptor2D, 668
 - mappel::MLEstimator, 717
 - mappel::PointEmitterModel, 781
- DefaultMCMCThin
 - mappel::Gauss1DMAP, 150
 - mappel::Gauss1DMLE, 179
 - mappel::Gauss1DModel, 207
 - mappel::Gauss1DsMAP, 236
 - mappel::Gauss1DsMLE, 266
 - mappel::Gauss1DsModel, 294
 - mappel::Gauss2DMAP, 325
 - mappel::Gauss2DMLE, 356
 - mappel::Gauss2DModel, 386
 - mappel::Gauss2DsMAP, 419
 - mappel::Gauss2DsMLE, 453
 - mappel::Gauss2DsModel, 485
 - mappel::Gauss2DsxyMAP, 514
 - mappel::Gauss2DsxyModel, 539
 - mappel::MAPEstimator, 603
 - mappel::MCMCAdaptor1Ds, 649
 - mappel::MCMCAdaptor1D, 630
 - mappel::MCMCAdaptor2Ds, 688
 - mappel::MCMCAdaptor2D, 668
 - mappel::MLEstimator, 717
 - mappel::PointEmitterModel, 781
- DefaultMaxSeqLength
 - mappel::estimator::IterativeMaximizer::Maximizer←
Data, 613
- DefaultNumIterations
 - mappel::estimator::SimulatedAnnealingMaximizer,
838
- DefaultPriorBetaPos
 - mappel::Gauss1DMAP, 150
 - mappel::Gauss1DMLE, 179
 - mappel::Gauss1DModel, 207
 - mappel::Gauss1DsMAP, 236
 - mappel::Gauss1DsMLE, 266
 - mappel::Gauss1DsModel, 294
 - mappel::Gauss2DMAP, 325
 - mappel::Gauss2DMLE, 356
 - mappel::Gauss2DModel, 386
 - mappel::Gauss2DsMAP, 419
 - mappel::Gauss2DsMLE, 453
 - mappel::Gauss2DsModel, 485
 - mappel::Gauss2DsxyMAP, 514
 - mappel::Gauss2DsxyModel, 539
 - mappel::MAPEstimator, 603
 - mappel::MCMCAdaptor1Ds, 649
 - mappel::MCMCAdaptor1D, 630
 - mappel::MCMCAdaptor2Ds, 688
 - mappel::MCMCAdaptor2D, 668
 - mappel::MLEstimator, 717
 - mappel::PointEmitterModel, 781
- DefaultPriorIntensityKappa
 - mappel::Gauss1DMAP, 150
 - mappel::Gauss1DMLE, 179
 - mappel::Gauss1DModel, 207
 - mappel::Gauss1DsMAP, 236
 - mappel::Gauss1DsMLE, 266
 - mappel::Gauss1DsModel, 295
 - mappel::Gauss2DMAP, 325
 - mappel::Gauss2DMLE, 356
 - mappel::Gauss2DModel, 387
 - mappel::Gauss2DsMAP, 419
 - mappel::Gauss2DsMLE, 453
 - mappel::Gauss2DsModel, 485
 - mappel::Gauss2DsxyMAP, 514
 - mappel::Gauss2DsxyModel, 539
 - mappel::MAPEstimator, 603
 - mappel::MCMCAdaptor1Ds, 649
 - mappel::MCMCAdaptor1D, 630
 - mappel::MCMCAdaptor2Ds, 688
 - mappel::MCMCAdaptor2D, 669
 - mappel::MLEstimator, 717
 - mappel::PointEmitterModel, 781
- DefaultPriorMaxI
 - mappel::Gauss1DMAP, 150
 - mappel::Gauss1DMLE, 179
 - mappel::Gauss1DModel, 207
 - mappel::Gauss1DsMAP, 237
 - mappel::Gauss1DsMLE, 266
 - mappel::Gauss1DsModel, 295
 - mappel::Gauss2DMAP, 325

- mappel::Gauss2DMLE, [356](#)
- mappel::Gauss2DModel, [387](#)
- mappel::Gauss2DsMAP, [420](#)
- mappel::Gauss2DsMLE, [453](#)
- mappel::Gauss2DsModel, [485](#)
- mappel::Gauss2DsxyMAP, [514](#)
- mappel::Gauss2DsxyModel, [539](#)
- mappel::MAPEstimator, [604](#)
- mappel::MCMCAdaptor1Ds, [649](#)
- mappel::MCMCAdaptor1D, [630](#)
- mappel::MCMCAdaptor2Ds, [688](#)
- mappel::MCMCAdaptor2D, [669](#)
- mappel::MLEstimator, [718](#)
- mappel::PointEmitterModel, [781](#)
- DefaultPriorMeanI
 - mappel::Gauss1DMAP, [150](#)
 - mappel::Gauss1DMLE, [179](#)
 - mappel::Gauss1DModel, [207](#)
 - mappel::Gauss1DsMAP, [237](#)
 - mappel::Gauss1DsMLE, [266](#)
 - mappel::Gauss1DsModel, [295](#)
 - mappel::Gauss2DMAP, [325](#)
 - mappel::Gauss2DMLE, [356](#)
 - mappel::Gauss2DModel, [387](#)
 - mappel::Gauss2DsMAP, [420](#)
 - mappel::Gauss2DsMLE, [453](#)
 - mappel::Gauss2DsModel, [485](#)
 - mappel::Gauss2DsxyMAP, [514](#)
 - mappel::Gauss2DsxyModel, [539](#)
 - mappel::MAPEstimator, [604](#)
 - mappel::MCMCAdaptor1Ds, [649](#)
 - mappel::MCMCAdaptor1D, [630](#)
 - mappel::MCMCAdaptor2Ds, [688](#)
 - mappel::MCMCAdaptor2D, [669](#)
 - mappel::MLEstimator, [718](#)
 - mappel::PointEmitterModel, [781](#)
- DefaultPriorPSFSigmaAlpha
 - mappel::Gauss1DMAP, [151](#)
 - mappel::Gauss1DMLE, [180](#)
 - mappel::Gauss1DModel, [208](#)
 - mappel::Gauss1DsMAP, [237](#)
 - mappel::Gauss1DsMLE, [266](#)
 - mappel::Gauss1DsModel, [295](#)
 - mappel::Gauss2DMAP, [326](#)
 - mappel::Gauss2DMLE, [357](#)
 - mappel::Gauss2DModel, [387](#)
 - mappel::Gauss2DsMAP, [420](#)
 - mappel::Gauss2DsMLE, [453](#)
 - mappel::Gauss2DsModel, [486](#)
 - mappel::Gauss2DsxyMAP, [514](#)
 - mappel::Gauss2DsxyModel, [540](#)
 - mappel::MAPEstimator, [604](#)
 - mappel::MCMCAdaptor1Ds, [650](#)
 - mappel::MCMCAdaptor1D, [631](#)
 - mappel::MCMCAdaptor2Ds, [689](#)
 - mappel::MCMCAdaptor2D, [669](#)
 - mappel::MLEstimator, [718](#)
 - mappel::PointEmitterModel, [782](#)
- mappel::MCMCAdaptor2Ds, [689](#)
- mappel::MCMCAdaptor2D, [669](#)
- mappel::MLEstimator, [718](#)
- mappel::PointEmitterModel, [782](#)
- DefaultPriorPixelMeanBG
 - mappel::Gauss1DMAP, [151](#)
 - mappel::Gauss1DMLE, [180](#)
 - mappel::Gauss1DModel, [208](#)
 - mappel::Gauss1DsMAP, [237](#)
 - mappel::Gauss1DsMLE, [266](#)
 - mappel::Gauss1DsModel, [295](#)
 - mappel::Gauss2DMAP, [325](#)
 - mappel::Gauss2DMLE, [356](#)
 - mappel::Gauss2DModel, [387](#)
 - mappel::Gauss2DsMAP, [420](#)
 - mappel::Gauss2DsMLE, [453](#)
 - mappel::Gauss2DsModel, [485](#)
 - mappel::Gauss2DsxyMAP, [514](#)
 - mappel::Gauss2DsxyModel, [540](#)
 - mappel::MAPEstimator, [604](#)
 - mappel::MCMCAdaptor1Ds, [650](#)
 - mappel::MCMCAdaptor1D, [630](#)
 - mappel::MCMCAdaptor2Ds, [689](#)
 - mappel::MCMCAdaptor2D, [669](#)
 - mappel::MLEstimator, [718](#)
 - mappel::PointEmitterModel, [781](#)
- DefaultPriorSigmaPos
 - mappel::Gauss1DMAP, [151](#)
 - mappel::Gauss1DMLE, [180](#)
 - mappel::Gauss1DModel, [208](#)
 - mappel::Gauss1DsMAP, [237](#)
 - mappel::Gauss1DsMLE, [267](#)
 - mappel::Gauss1DsModel, [295](#)
 - mappel::Gauss2DMAP, [326](#)
 - mappel::Gauss2DMLE, [357](#)
 - mappel::Gauss2DModel, [387](#)
 - mappel::Gauss2DsMAP, [420](#)
 - mappel::Gauss2DsMLE, [454](#)
 - mappel::Gauss2DsModel, [486](#)
 - mappel::Gauss2DsxyMAP, [515](#)
 - mappel::Gauss2DsxyModel, [540](#)
 - mappel::MAPEstimator, [604](#)
 - mappel::MCMCAdaptor1Ds, [650](#)
 - mappel::MCMCAdaptor1D, [631](#)
 - mappel::MCMCAdaptor2Ds, [689](#)
 - mappel::MCMCAdaptor2D, [669](#)
 - mappel::MLEstimator, [718](#)
 - mappel::PointEmitterModel, [782](#)
- DefaultPriorType
 - mappel::Gauss1DMAP, [151](#)
 - mappel::Gauss1DMLE, [180](#)
 - mappel::Gauss1DModel, [208](#)
 - mappel::Gauss1DsMAP, [237](#)
 - mappel::Gauss1DsMLE, [267](#)

- mappel::Gauss1DsModel, [295](#)
- mappel::Gauss2DModel, [326](#)
- mappel::Gauss2DModel, [357](#)
- mappel::Gauss2DModel, [387](#)
- mappel::Gauss2DsMAP, [420](#)
- mappel::Gauss2DsMLE, [454](#)
- mappel::Gauss2DsModel, [486](#)
- mappel::Gauss2DsxyMAP, [515](#)
- DefaultProfileBoundsEstimatorMethod
 - mappel::Gauss1DModel, [151](#)
 - mappel::Gauss1DModel, [180](#)
 - mappel::Gauss1DModel, [208](#)
 - mappel::Gauss1DsMAP, [238](#)
 - mappel::Gauss1DsMLE, [267](#)
 - mappel::Gauss1DsModel, [296](#)
 - mappel::Gauss2DModel, [326](#)
 - mappel::Gauss2DModel, [357](#)
 - mappel::Gauss2DModel, [388](#)
 - mappel::Gauss2DsMAP, [421](#)
 - mappel::Gauss2DsMLE, [454](#)
 - mappel::Gauss2DsModel, [486](#)
 - mappel::Gauss2DsxyMAP, [515](#)
 - mappel::Gauss2DsxyModel, [540](#)
 - mappel::MAPEstimator, [604](#)
 - mappel::MCMCAdaptor1Ds, [650](#)
 - mappel::MCMCAdaptor1D, [631](#)
 - mappel::MCMCAdaptor2Ds, [689](#)
 - mappel::MCMCAdaptor2D, [670](#)
 - mappel::MLEstimator, [718](#)
 - mappel::PointEmitterModel, [782](#)
- DefaultSeperableInitEstimator
 - mappel::Gauss1DModel, [151](#)
 - mappel::Gauss1DModel, [180](#)
 - mappel::Gauss1DModel, [208](#)
 - mappel::Gauss1DsMAP, [238](#)
 - mappel::Gauss1DsMLE, [267](#)
 - mappel::Gauss1DsModel, [296](#)
 - mappel::Gauss2DModel, [326](#)
 - mappel::Gauss2DModel, [357](#)
 - mappel::Gauss2DModel, [388](#)
 - mappel::Gauss2DsMAP, [421](#)
 - mappel::Gauss2DsMLE, [454](#)
 - mappel::Gauss2DsModel, [486](#)
 - mappel::Gauss2DsxyMAP, [515](#)
 - mappel::Gauss2DsxyModel, [540](#)
 - mappel::MAPEstimator, [605](#)
 - mappel::MCMCAdaptor1Ds, [650](#)
 - mappel::MCMCAdaptor1D, [631](#)
 - mappel::MCMCAdaptor2Ds, [689](#)
 - mappel::MCMCAdaptor2D, [670](#)
 - mappel::MLEstimator, [719](#)
 - mappel::PointEmitterModel, [782](#)
- derivatives_computed
 - mappel::Gauss1DModel::Stencil, [862](#)
 - mappel::Gauss1DsModel::Stencil, [848](#)
 - mappel::Gauss2DModel::Stencil, [852](#)
 - mappel::Gauss2DsModel::Stencil, [857](#)
 - mappel::Gauss2DsxyModel::Stencil, [843](#)
- display.cpp, [899](#)
- display.h, [900](#)
- DoNotTry
 - omp_exception_catcher, [90](#)
- DX
 - mappel::Gauss1DModel::Stencil, [862](#)
 - mappel::Gauss1DsModel::Stencil, [848](#)
 - mappel::Gauss2DModel::Stencil, [852](#)
 - mappel::Gauss2DsModel::Stencil, [857](#)
 - mappel::Gauss2DsxyModel::Stencil, [843](#)
- dx
 - mappel::Gauss1DModel::Stencil, [862](#)
 - mappel::Gauss1DsModel::Stencil, [848](#)
 - mappel::Gauss2DModel::Stencil, [852](#)
 - mappel::Gauss2DsModel::Stencil, [857](#)
 - mappel::Gauss2DsxyModel::Stencil, [843](#)
- DY
 - mappel::Gauss2DModel::Stencil, [853](#)
 - mappel::Gauss2DsModel::Stencil, [858](#)
 - mappel::Gauss2DsxyModel::Stencil, [844](#)
- dy
 - mappel::Gauss2DModel::Stencil, [852](#)
 - mappel::Gauss2DsModel::Stencil, [858](#)
 - mappel::Gauss2DsxyModel::Stencil, [844](#)
- 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, [85](#)
- error_bounds_observed
 - mappel::methods, [59](#)
- error_bounds_observed_stack
 - mappel::methods::openmp, [85](#)
- error_bounds_posterior_credible
 - mappel::methods, [59](#)
- error_bounds_profile_likelihood
 - mappel::methods, [59](#), [60](#)
- error_bounds_profile_likelihood_debug
 - mappel::methods::debug, [67](#)
- error_bounds_profile_likelihood_parallel
 - mappel::methods::openmp, [85](#)
- error_bounds_profile_likelihood_stack
 - mappel::methods::openmp, [86](#)
- estimate_background

- mappel, 33, 34
- estimate_gaussian_2Dmax
 - mappel, 34
- estimate_gaussian_3Dmax
 - mappel, 34
- estimate_intensity
 - mappel, 34
- estimate_max
 - mappel::estimator::CGaussHeuristicEstimator, 94, 95
 - mappel::estimator::CGaussMLE, 106, 107
 - mappel::estimator::Estimator, 120, 121
 - mappel::estimator::HeuristicEstimator, 547, 548
 - mappel::estimator::IterativeMaximizer, 575, 576
 - mappel::estimator::NewtonDiagonalMaximizer, 727, 728
 - mappel::estimator::NewtonMaximizer, 748, 749
 - mappel::estimator::QuasiNewtonMaximizer, 813, 814
 - mappel::estimator::SimulatedAnnealingMaximizer, 831, 832
 - mappel::estimator::ThreadedEstimator, 867, 868
 - mappel::estimator::TrustRegionMaximizer, 882, 883
 - mappel::methods, 60
- estimate_max_debug
 - mappel::estimator::CGaussHeuristicEstimator, 96
 - mappel::estimator::CGaussMLE, 108
 - mappel::estimator::Estimator, 121, 122
 - mappel::estimator::HeuristicEstimator, 549
 - mappel::estimator::IterativeMaximizer, 576, 577
 - mappel::estimator::NewtonDiagonalMaximizer, 729
 - mappel::estimator::NewtonMaximizer, 750
 - mappel::estimator::QuasiNewtonMaximizer, 815
 - mappel::estimator::SimulatedAnnealingMaximizer, 832, 833
 - mappel::estimator::ThreadedEstimator, 868, 869
 - mappel::estimator::TrustRegionMaximizer, 884
 - mappel::methods::debug, 67
- estimate_max_stack
 - mappel::estimator::CGaussHeuristicEstimator, 97
 - mappel::estimator::CGaussMLE, 109
 - mappel::estimator::Estimator, 122, 123
 - mappel::estimator::HeuristicEstimator, 550
 - mappel::estimator::IterativeMaximizer, 577, 578
 - mappel::estimator::NewtonDiagonalMaximizer, 730
 - mappel::estimator::NewtonMaximizer, 751
 - mappel::estimator::QuasiNewtonMaximizer, 816
 - mappel::estimator::SimulatedAnnealingMaximizer, 833, 834
 - mappel::estimator::ThreadedEstimator, 869, 870
 - mappel::estimator::TrustRegionMaximizer, 885
 - mappel::methods::openmp, 86
- estimate_posterior
 - mappel::methods, 60, 61
- estimate_posterior_debug
 - mappel::methods::debug, 67
- estimate_posterior_stack
 - mappel::methods::openmp, 87
- estimate_profile_bounds
 - mappel::estimator::CGaussHeuristicEstimator, 98
 - mappel::estimator::CGaussMLE, 110
 - mappel::estimator::Estimator, 123
 - mappel::estimator::HeuristicEstimator, 551
 - mappel::estimator::IterativeMaximizer, 578
 - mappel::estimator::NewtonDiagonalMaximizer, 731
 - mappel::estimator::NewtonMaximizer, 752
 - mappel::estimator::QuasiNewtonMaximizer, 817
 - mappel::estimator::SimulatedAnnealingMaximizer, 834
 - mappel::estimator::ThreadedEstimator, 870
 - mappel::estimator::TrustRegionMaximizer, 886
- estimate_profile_bounds_debug
 - mappel::estimator::CGaussHeuristicEstimator, 98
 - mappel::estimator::CGaussMLE, 110
 - mappel::estimator::Estimator, 123
 - mappel::estimator::HeuristicEstimator, 551
 - mappel::estimator::IterativeMaximizer, 579
 - mappel::estimator::NewtonDiagonalMaximizer, 731
 - mappel::estimator::NewtonMaximizer, 752
 - mappel::estimator::QuasiNewtonMaximizer, 817
 - mappel::estimator::SimulatedAnnealingMaximizer, 835
 - mappel::estimator::ThreadedEstimator, 870
 - mappel::estimator::TrustRegionMaximizer, 886
- estimate_profile_bounds_parallel
 - mappel::estimator::CGaussHeuristicEstimator, 99
 - mappel::estimator::CGaussMLE, 111
 - mappel::estimator::Estimator, 124
 - mappel::estimator::HeuristicEstimator, 551
 - mappel::estimator::IterativeMaximizer, 579
 - mappel::estimator::NewtonDiagonalMaximizer, 731
 - mappel::estimator::NewtonMaximizer, 752
 - mappel::estimator::QuasiNewtonMaximizer, 817
 - mappel::estimator::SimulatedAnnealingMaximizer, 835
 - mappel::estimator::ThreadedEstimator, 871
 - mappel::estimator::TrustRegionMaximizer, 887
- estimate_profile_bounds_stack
 - mappel::estimator::CGaussHeuristicEstimator, 99
 - mappel::estimator::CGaussMLE, 111
 - mappel::estimator::Estimator, 124
 - mappel::estimator::HeuristicEstimator, 552
 - mappel::estimator::IterativeMaximizer, 579
 - mappel::estimator::NewtonDiagonalMaximizer, 731
 - mappel::estimator::NewtonMaximizer, 752
 - mappel::estimator::QuasiNewtonMaximizer, 817
 - mappel::estimator::SimulatedAnnealingMaximizer, 835
 - mappel::estimator::ThreadedEstimator, 871

- mappel::estimator::TrustRegionMaximizer, 887
- estimate_profile_likelihood
 - mappel::methods, 61
- estimate_profile_likelihood_stack
 - mappel::methods::openmp, 87
- estimate_profile_max
 - mappel::estimator::CGaussHeuristicEstimator, 100
 - mappel::estimator::CGaussMLE, 112
 - mappel::estimator::Estimator, 124
 - mappel::estimator::HeuristicEstimator, 552, 553
 - mappel::estimator::IterativeMaximizer, 580
 - mappel::estimator::NewtonDiagonalMaximizer, 732
 - mappel::estimator::NewtonMaximizer, 753
 - mappel::estimator::QuasiNewtonMaximizer, 818
 - mappel::estimator::SimulatedAnnealingMaximizer, 836
 - mappel::estimator::ThreadedEstimator, 872
 - mappel::estimator::TrustRegionMaximizer, 888
- estimate_sample_posterior
 - mappel::mcmc, 54
- estimated_idx
 - mappel::estimator::ProfileBoundsData, 802
 - mappel::estimator::ProfileBoundsDataStack, 805
- Estimator
 - mappel::estimator::Estimator, 118
- estimator.cpp, 901
- estimator.h, 902
 - ClockT, 905
- estimator_helpers.h, 905
- estimator_impl.h, 906
- estimator_names
 - mappel::Gauss1DMAP, 152
 - mappel::Gauss1DMLE, 181
 - mappel::Gauss1DsMAP, 238
 - mappel::Gauss1DsMLE, 267
 - mappel::Gauss2DMAP, 326
 - mappel::Gauss2DMLE, 357
 - mappel::Gauss2DsMAP, 421
 - mappel::Gauss2DsMLE, 454
 - mappel::Gauss2DsxyMAP, 515
 - mappel::PoissonNoise1DObjective, 792
 - mappel::PoissonNoise2DObjective, 799
 - PoissonGaussianNoise2DObjective, 785
- estimator_statics.cpp, 907
- eta_bg
 - mappel::Gauss1DMAP, 152
 - mappel::Gauss1DMLE, 181
 - mappel::Gauss1DModel, 209
 - mappel::Gauss1DsMAP, 238
 - mappel::Gauss1DsMLE, 267
 - mappel::Gauss1DsModel, 296
 - mappel::Gauss2DMAP, 327
 - mappel::Gauss2DMLE, 358
 - mappel::Gauss2DModel, 388
 - mappel::Gauss2DsMAP, 422
 - mappel::Gauss2DsMLE, 455
 - mappel::Gauss2DsModel, 487
 - mappel::Gauss2DsxyMAP, 516
 - mappel::MCMCAdaptor1Ds, 651
 - mappel::MCMCAdaptor1D, 632
 - mappel::MCMCAdaptor2Ds, 690
 - mappel::MCMCAdaptor2D, 670
- eta_l
 - mappel::Gauss1DMAP, 152
 - mappel::Gauss1DMLE, 181
 - mappel::Gauss1DModel, 209
 - mappel::Gauss1DsMAP, 238
 - mappel::Gauss1DsMLE, 267
 - mappel::Gauss1DsModel, 296
 - mappel::Gauss2DMAP, 327
 - mappel::Gauss2DMLE, 358
 - mappel::Gauss2DModel, 388
 - mappel::Gauss2DsMAP, 421
 - mappel::Gauss2DsMLE, 454
 - mappel::Gauss2DsModel, 487
 - mappel::Gauss2DsxyMAP, 515
 - mappel::MCMCAdaptor1Ds, 651
 - mappel::MCMCAdaptor1D, 631
 - mappel::MCMCAdaptor2Ds, 690
 - mappel::MCMCAdaptor2D, 670
- eta_sigma
 - mappel::Gauss1DsMAP, 238
 - mappel::Gauss1DsMLE, 268
 - mappel::Gauss1DsModel, 296
 - mappel::Gauss2DsMAP, 421
 - mappel::Gauss2DsMLE, 455
 - mappel::Gauss2DsModel, 487
 - mappel::MCMCAdaptor1Ds, 651
 - mappel::MCMCAdaptor2Ds, 690
- eta_x
 - mappel::Gauss1DMAP, 152
 - mappel::Gauss1DMLE, 181
 - mappel::Gauss1DModel, 209
 - mappel::Gauss1DsMAP, 239
 - mappel::Gauss1DsMLE, 268
 - mappel::Gauss1DsModel, 296
 - mappel::Gauss2DMAP, 327
 - mappel::Gauss2DMLE, 358
 - mappel::Gauss2DModel, 388
 - mappel::Gauss2DsMAP, 422
 - mappel::Gauss2DsMLE, 455
 - mappel::Gauss2DsModel, 487
 - mappel::Gauss2DsxyMAP, 516
 - mappel::MCMCAdaptor1Ds, 651
 - mappel::MCMCAdaptor1D, 632
 - mappel::MCMCAdaptor2Ds, 690
 - mappel::MCMCAdaptor2D, 670
- eta_y

- mappel::Gauss2DModel, 388
- mappel::Gauss2DsMAP, 422
- mappel::Gauss2DsMLE, 455
- mappel::Gauss2DsModel, 487
- mappel::MCMCAdaptor2Ds, 690
- mappel::MCMCAdaptor2D, 670
- exit_counts
 - mappel::estimator::CGaussHeuristicEstimator, 102
 - mappel::estimator::CGaussMLE, 114
 - mappel::estimator::Estimator, 126
 - mappel::estimator::HeuristicEstimator, 555
 - mappel::estimator::IterativeMaximizer, 586
 - mappel::estimator::NewtonDiagonalMaximizer, 739
 - mappel::estimator::NewtonMaximizer, 760
 - mappel::estimator::QuasiNewtonMaximizer, 825
 - mappel::estimator::SimulatedAnnealingMaximizer, 839
 - mappel::estimator::ThreadedEstimator, 874
 - mappel::estimator::TrustRegionMaximizer, 895
- ExitCode
 - mappel::estimator, 49
- expand_max_seq_len
 - mappel::estimator::IterativeMaximizer::Maximizer←Data, 609
- expected_information
 - mappel::methods, 61, 62
- expected_information_stack
 - mappel::methods::openmp, 87
- 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, 953
- fixed_idx
 - mappel::estimator::IterativeMaximizer::Maximizer←Data, 613
- free_idx
 - mappel::estimator::IterativeMaximizer::Maximizer←Data, 613
- FunctionValue
 - mappel::estimator, 50
- gamma_prior_grad
 - mappel, 35
- gamma_prior_grad2
 - mappel, 35
- Gauss1DModel.cpp, 907
- Gauss1DModel.h, 908
- Gauss1DModel
 - mappel::Gauss1DModel, 134
- Gauss1DMLE.cpp, 908
- Gauss1DMLE.h, 909
- Gauss1DMLE
 - mappel::Gauss1DMLE, 163
- Gauss1DModel
 - mappel::Gauss1DModel, 191
- Gauss1DModel.cpp, 910
- Gauss1DModel.h, 910
- Gauss1DSumModelT
 - mappel::Gauss2DModel, 305
 - mappel::Gauss2DMLE, 337
 - mappel::Gauss2DModel, 368
 - mappel::Gauss2DsMAP, 398
 - mappel::Gauss2DsMLE, 432
 - mappel::Gauss2DsModel, 465
- Gauss1DsMAP.cpp, 911
- Gauss1DsMAP.h, 911
- Gauss1DsMAP
 - mappel::Gauss1DsMAP, 219, 220
- Gauss1DsMLE.cpp, 912
- Gauss1DsMLE.h, 913
- Gauss1DsMLE
 - mappel::Gauss1DsMLE, 249
- Gauss1DsModel
 - mappel::Gauss1DsModel, 278
- Gauss1DsModel.cpp, 913
- Gauss1DsModel.h, 914
- Gauss2DModel.cpp, 915
- Gauss2DModel.h, 915
- Gauss2DModel
 - mappel::Gauss2DModel, 307, 308
- Gauss2DMLE.cpp, 916
- Gauss2DMLE.h, 916
- Gauss2DMLE
 - mappel::Gauss2DMLE, 339
- Gauss2DModel
 - mappel::Gauss2DModel, 370
- Gauss2DModel.cpp, 917
- Gauss2DModel.h, 918
- Gauss2DsMAP.cpp, 919
- Gauss2DsMAP.h, 919

- Gauss2DsMAP
 - mappel::Gauss2DsMAP, [400](#), [401](#)
- Gauss2DsMLE.cpp, [920](#)
- Gauss2DsMLE.h, [920](#)
- Gauss2DsMLE
 - mappel::Gauss2DsMLE, [434](#)
- Gauss2DsModel
 - mappel::Gauss2DsModel, [467](#)
- Gauss2DsModel.cpp, [921](#)
- Gauss2DsModel.h, [922](#)
- Gauss2DsxyMAP.h, [923](#)
- Gauss2DsxyMAP
 - mappel::Gauss2DsxyMAP, [498](#)
- Gauss2DsxyModel
 - mappel::Gauss2DsxyModel, [526](#)
- Gauss2DsxyModel.h, [923](#)
- 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, [609](#)
- get_debug_stats
 - mappel::estimator::CGaussHeuristicEstimator, [100](#)
 - mappel::estimator::CGaussMLE, [112](#)
 - mappel::estimator::Estimator, [124](#)
 - mappel::estimator::HeuristicEstimator, [553](#)
 - mappel::estimator::IterativeMaximizer, [580](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [732](#)
 - mappel::estimator::NewtonMaximizer, [753](#)
 - mappel::estimator::QuasiNewtonMaximizer, [818](#)
 - mappel::estimator::SimulatedAnnealingMaximizer, [836](#)
 - mappel::estimator::ThreadedEstimator, [872](#)
 - mappel::estimator::TrustRegionMaximizer, [888](#)
- get_exit_counts
 - mappel::estimator::CGaussHeuristicEstimator, [100](#)
 - mappel::estimator::CGaussMLE, [113](#)
 - mappel::estimator::Estimator, [125](#)
 - mappel::estimator::HeuristicEstimator, [553](#)
 - mappel::estimator::IterativeMaximizer, [581](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [733](#)
 - mappel::estimator::NewtonMaximizer, [754](#)
 - mappel::estimator::QuasiNewtonMaximizer, [819](#)
 - mappel::estimator::SimulatedAnnealingMaximizer, [837](#)
 - mappel::estimator::ThreadedEstimator, [873](#)
 - mappel::estimator::TrustRegionMaximizer, [889](#)
- get_hyperparam_index
 - mappel::Gauss1DMAP, [136](#)
 - mappel::Gauss1DMLE, [165](#)
 - mappel::Gauss1DModel, [193](#)
 - mappel::Gauss1DsMAP, [222](#)
 - mappel::Gauss1DsMLE, [251](#)
 - mappel::Gauss1DsModel, [280](#)
 - mappel::Gauss2DMAP, [310](#)
 - mappel::Gauss2DMLE, [341](#)
 - mappel::Gauss2DModel, [372](#)
 - mappel::Gauss2DsMAP, [403](#)
 - mappel::Gauss2DsMLE, [437](#)
 - mappel::Gauss2DsModel, [470](#)
 - mappel::Gauss2DsxyMAP, [500](#)
 - mappel::Gauss2DsxyModel, [528](#)
 - mappel::MAPEstimator, [595](#)
 - mappel::MCMCAdaptor1Ds, [639](#)
 - mappel::MCMCAdaptor1D, [620](#)
 - mappel::MCMCAdaptor2Ds, [678](#)
 - mappel::MCMCAdaptor2D, [659](#)
 - mappel::MLEstimator, [709](#)
 - mappel::PointEmitterModel, [773](#)
- get_hyperparam_names
 - mappel::Gauss1DMAP, [136](#)
 - mappel::Gauss1DMLE, [165](#)
 - mappel::Gauss1DModel, [193](#)
 - mappel::Gauss1DsMAP, [222](#)
 - mappel::Gauss1DsMLE, [251](#)
 - mappel::Gauss1DsModel, [280](#)
 - mappel::Gauss2DMAP, [310](#)
 - mappel::Gauss2DMLE, [341](#)
 - mappel::Gauss2DModel, [372](#)
 - mappel::Gauss2DsMAP, [403](#)
 - mappel::Gauss2DsMLE, [437](#)
 - mappel::Gauss2DsModel, [470](#)
 - mappel::Gauss2DsxyMAP, [500](#)
 - mappel::Gauss2DsxyModel, [528](#)
 - mappel::MAPEstimator, [595](#)
 - mappel::MCMCAdaptor1Ds, [639](#)
 - mappel::MCMCAdaptor1D, [620](#)
 - mappel::MCMCAdaptor2Ds, [679](#)
 - mappel::MCMCAdaptor2D, [659](#)
 - mappel::MLEstimator, [709](#)
 - mappel::PointEmitterModel, [773](#)
- get_hyperparam_value
 - mappel::Gauss1DMAP, [136](#)
 - mappel::Gauss1DMLE, [165](#)
 - mappel::Gauss1DModel, [194](#)
 - mappel::Gauss1DsMAP, [222](#)
 - mappel::Gauss1DsMLE, [251](#)

- mappel::Gauss1DsModel, [280](#)
- mappel::Gauss2DMAP, [310](#)
- mappel::Gauss2DMLE, [342](#)
- mappel::Gauss2DModel, [372](#)
- mappel::Gauss2DsMAP, [404](#)
- mappel::Gauss2DsMLE, [437](#)
- mappel::Gauss2DsModel, [470](#)
- mappel::Gauss2DsxyMAP, [500](#)
- mappel::Gauss2DsxyModel, [528](#)
- mappel::MAPEstimator, [595](#)
- mappel::MCMCAdaptor1Ds, [640](#)
- mappel::MCMCAdaptor1D, [621](#)
- mappel::MCMCAdaptor2Ds, [679](#)
- mappel::MCMCAdaptor2D, [659](#)
- mappel::MLEstimator, [709](#)
- mappel::PointEmitterModel, [773](#)
- get_hyperparams
 - mappel::Gauss1DMAP, [136](#)
 - mappel::Gauss1DMLE, [165](#)
 - mappel::Gauss1DModel, [194](#)
 - mappel::Gauss1DsMAP, [222](#)
 - mappel::Gauss1DsMLE, [251](#)
 - mappel::Gauss1DsModel, [280](#)
 - mappel::Gauss2DMAP, [311](#)
 - mappel::Gauss2DMLE, [342](#)
 - mappel::Gauss2DModel, [372](#)
 - mappel::Gauss2DsMAP, [404](#)
 - mappel::Gauss2DsMLE, [437](#)
 - mappel::Gauss2DsModel, [470](#)
 - mappel::Gauss2DsxyMAP, [500](#)
 - mappel::Gauss2DsxyModel, [528](#)
 - mappel::MAPEstimator, [595](#)
 - mappel::MCMCAdaptor1Ds, [640](#)
 - mappel::MCMCAdaptor1D, [621](#)
 - mappel::MCMCAdaptor2Ds, [679](#)
 - mappel::MCMCAdaptor2D, [659](#)
 - mappel::MLEstimator, [709](#)
 - mappel::PointEmitterModel, [773](#)
- get_image_from_stack
 - mappel::Gauss1DMAP, [137](#)
 - mappel::Gauss1DMLE, [166](#)
 - mappel::Gauss1DModel, [194](#)
 - mappel::Gauss1DsMAP, [222](#)
 - mappel::Gauss1DsMLE, [252](#)
 - mappel::Gauss1DsModel, [281](#)
 - mappel::Gauss2DMAP, [311](#)
 - mappel::Gauss2DMLE, [342](#)
 - mappel::Gauss2DModel, [373](#)
 - mappel::Gauss2DsMAP, [404](#)
 - mappel::Gauss2DsMLE, [438](#)
 - mappel::Gauss2DsModel, [470](#)
 - mappel::Gauss2DsxyMAP, [501](#)
 - mappel::Gauss2DsxyModel, [528](#)
 - mappel::ImageFormat1DBase, [560](#)
- mappel::ImageFormat2DBase, [566](#)
- mappel::PoissonNoise1DObjective, [790](#)
- mappel::PoissonNoise2DObjective, [797](#)
- get_lbound
 - mappel::Gauss1DMAP, [137](#)
 - mappel::Gauss1DMLE, [166](#)
 - mappel::Gauss1DModel, [194](#)
 - mappel::Gauss1DsMAP, [222](#)
 - mappel::Gauss1DsMLE, [252](#)
 - mappel::Gauss1DsModel, [281](#)
 - mappel::Gauss2DMAP, [311](#)
 - mappel::Gauss2DMLE, [342](#)
 - mappel::Gauss2DModel, [373](#)
 - mappel::Gauss2DsMAP, [404](#)
 - mappel::Gauss2DsMLE, [438](#)
 - mappel::Gauss2DsModel, [470](#)
 - mappel::Gauss2DsxyMAP, [501](#)
 - mappel::Gauss2DsxyModel, [528](#)
 - mappel::MAPEstimator, [596](#)
 - mappel::MCMCAdaptor1Ds, [640](#)
 - mappel::MCMCAdaptor1D, [621](#)
 - mappel::MCMCAdaptor2Ds, [679](#)
 - mappel::MCMCAdaptor2D, [659](#)
 - mappel::MLEstimator, [709](#)
 - mappel::PointEmitterModel, [773](#)
- get_max_sigma
 - mappel::Gauss1DsMAP, [223](#)
 - mappel::Gauss1DsMLE, [252](#)
 - mappel::Gauss1DsModel, [281](#)
 - mappel::Gauss2DsMAP, [404](#)
 - mappel::Gauss2DsMLE, [438](#)
 - mappel::Gauss2DsModel, [470](#), [471](#)
 - mappel::Gauss2DsxyModel, [529](#)
- get_max_sigma_ratio
 - mappel::Gauss2DsMAP, [405](#)
 - mappel::Gauss2DsMLE, [438](#)
 - mappel::Gauss2DsModel, [471](#)
 - mappel::Gauss2DsxyModel, [529](#)
- get_mcmc_num_phases
 - mappel::Gauss1DMAP, [137](#)
 - mappel::Gauss1DMLE, [166](#)
 - mappel::Gauss1DModel, [194](#)
 - mappel::Gauss1DsMAP, [223](#)
 - mappel::Gauss1DsMLE, [252](#)
 - mappel::Gauss1DsModel, [281](#)
 - mappel::Gauss2DMAP, [311](#)
 - mappel::Gauss2DMLE, [342](#)
 - mappel::Gauss2DModel, [373](#)
 - mappel::Gauss2DsMAP, [405](#)
 - mappel::Gauss2DsMLE, [438](#)
 - mappel::Gauss2DsModel, [471](#)
 - mappel::Gauss2DsxyMAP, [501](#)
 - mappel::MCMCAdaptor1Ds, [640](#)
 - mappel::MCMCAdaptor1D, [621](#)

- mappel::MCMCAdaptor2Ds, 679
- mappel::MCMCAdaptor2D, 660
- mappel::MCMCAdaptorBase, 694
- get_mcmc_sigma_scale
 - mappel::Gauss1DMAP, 137
 - mappel::Gauss1DMLE, 166
 - mappel::Gauss1DModel, 194
 - mappel::Gauss1DsMAP, 223
 - mappel::Gauss1DsMLE, 252
 - mappel::Gauss1DsModel, 281
 - mappel::Gauss2DMAP, 311
 - mappel::Gauss2DMLE, 342
 - mappel::Gauss2DModel, 373
 - mappel::Gauss2DsMAP, 405
 - mappel::Gauss2DsMLE, 438
 - mappel::Gauss2DsModel, 471
 - mappel::Gauss2DsxyMAP, 501
 - mappel::MCMCAdaptor1Ds, 640
 - mappel::MCMCAdaptor1D, 621
 - mappel::MCMCAdaptor2Ds, 679
 - mappel::MCMCAdaptor2D, 660
 - mappel::MCMCAdaptorBase, 694
- get_min_sigma
 - mappel::Gauss1DsMAP, 223
 - mappel::Gauss1DsMLE, 252
 - mappel::Gauss1DsModel, 281
 - mappel::Gauss2DsMAP, 405
 - mappel::Gauss2DsMLE, 439
 - mappel::Gauss2DsModel, 471
 - mappel::Gauss2DsxyModel, 529
- get_model
 - mappel::estimator::CGaussHeuristicEstimator, 101
 - mappel::estimator::CGaussMLE, 113
 - mappel::estimator::Estimator, 125
 - mappel::estimator::HeuristicEstimator, 553
 - mappel::estimator::IterativeMaximizer, 581
 - mappel::estimator::NewtonDiagonalMaximizer, 733
 - mappel::estimator::NewtonMaximizer, 754
 - mappel::estimator::QuasiNewtonMaximizer, 819
 - mappel::estimator::SimulatedAnnealingMaximizer, 837
 - mappel::estimator::ThreadedEstimator, 873
 - mappel::estimator::TrustRegionMaximizer, 889
- get_num_hyperparams
 - mappel::Gauss1DMAP, 137
 - mappel::Gauss1DMLE, 166
 - mappel::Gauss1DModel, 194
 - mappel::Gauss1DsMAP, 223
 - mappel::Gauss1DsMLE, 252
 - mappel::Gauss1DsModel, 281
 - mappel::Gauss2DMAP, 311
 - mappel::Gauss2DMLE, 342
 - mappel::Gauss2DModel, 373
 - mappel::Gauss2DsMAP, 405
 - mappel::Gauss2DsMLE, 439
 - mappel::Gauss2DsModel, 472
 - mappel::Gauss2DsxyMAP, 501
 - mappel::Gauss2DsxyModel, 529
 - mappel::ImageFormat1DBase, 560
 - mappel::ImageFormat2DBase, 567
 - mappel::PoissonNoise1DObjective, 790
 - mappel::PoissonNoise2DObjective, 798
- mappel::Gauss2DsMLE, 439
- mappel::Gauss2DsModel, 471
- mappel::Gauss2DsxyMAP, 501
- mappel::Gauss2DsxyModel, 529
- mappel::MAPEstimator, 596
- mappel::MCMCAdaptor1Ds, 640
- mappel::MCMCAdaptor1D, 621
- mappel::MCMCAdaptor2Ds, 679
- mappel::MCMCAdaptor2D, 660
- mappel::MLEstimator, 710
- mappel::PointEmitterModel, 773
- get_num_params
 - mappel::Gauss1DMAP, 137
 - mappel::Gauss1DMLE, 166
 - mappel::Gauss1DModel, 195
 - mappel::Gauss1DsMAP, 223
 - mappel::Gauss1DsMLE, 253
 - mappel::Gauss1DsModel, 282
 - mappel::Gauss2DMAP, 311
 - mappel::Gauss2DMLE, 343
 - mappel::Gauss2DModel, 373
 - mappel::Gauss2DsMAP, 405
 - mappel::Gauss2DsMLE, 439
 - mappel::Gauss2DsModel, 472
 - mappel::Gauss2DsxyMAP, 501
 - mappel::Gauss2DsxyModel, 529
 - mappel::MAPEstimator, 596
 - mappel::MCMCAdaptor1Ds, 640
 - mappel::MCMCAdaptor1D, 621
 - mappel::MCMCAdaptor2Ds, 680
 - mappel::MCMCAdaptor2D, 660
 - mappel::MLEstimator, 710
 - mappel::PointEmitterModel, 774
- get_num_pixels
 - mappel::Gauss1DMAP, 137
 - mappel::Gauss1DMLE, 166
 - mappel::Gauss1DModel, 195
 - mappel::Gauss1DsMAP, 223
 - mappel::Gauss1DsMLE, 253
 - mappel::Gauss1DsModel, 282
 - mappel::Gauss2DMAP, 312
 - mappel::Gauss2DMLE, 343
 - mappel::Gauss2DModel, 373
 - mappel::Gauss2DsMAP, 406
 - mappel::Gauss2DsMLE, 439
 - mappel::Gauss2DsModel, 472
 - mappel::Gauss2DsxyMAP, 501
 - mappel::Gauss2DsxyModel, 529
 - mappel::ImageFormat1DBase, 560
 - mappel::ImageFormat2DBase, 567
 - mappel::PoissonNoise1DObjective, 790
 - mappel::PoissonNoise2DObjective, 798
- get_param_names
 - mappel::Gauss1DMAP, 138

- mappel::Gauss1DMLE, 167
- mappel::Gauss1DModel, 195
- mappel::Gauss1DsMAP, 224
- mappel::Gauss1DsMLE, 253
- mappel::Gauss1DsModel, 282
- mappel::Gauss2DMAP, 312
- mappel::Gauss2DMLE, 343
- mappel::Gauss2DModel, 374
- mappel::Gauss2DsMAP, 406
- mappel::Gauss2DsMLE, 439
- mappel::Gauss2DsModel, 472
- mappel::Gauss2DsxyMAP, 502
- mappel::Gauss2DsxyModel, 530
- mappel::MAPEstimator, 596
- mappel::MCMCAdaptor1Ds, 641
- mappel::MCMCAdaptor1D, 622
- mappel::MCMCAdaptor2Ds, 680
- mappel::MCMCAdaptor2D, 660
- mappel::MLEstimator, 710
- mappel::PointEmitterModel, 774
- get_prior
 - mappel::Gauss1DMAP, 138
 - mappel::Gauss1DMLE, 167
 - mappel::Gauss1DModel, 195
 - mappel::Gauss1DsMAP, 224
 - mappel::Gauss1DsMLE, 253
 - mappel::Gauss1DsModel, 282
 - mappel::Gauss2DMAP, 312
 - mappel::Gauss2DMLE, 343
 - mappel::Gauss2DModel, 374
 - mappel::Gauss2DsMAP, 406
 - mappel::Gauss2DsMLE, 439, 440
 - mappel::Gauss2DsModel, 472
 - mappel::Gauss2DsxyMAP, 502
 - mappel::Gauss2DsxyModel, 530
 - mappel::MAPEstimator, 596
 - mappel::MCMCAdaptor1Ds, 641
 - mappel::MCMCAdaptor1D, 622
 - mappel::MCMCAdaptor2Ds, 680
 - mappel::MCMCAdaptor2D, 660
 - mappel::MLEstimator, 710
 - mappel::PointEmitterModel, 774
- get_psf_sigma
 - mappel::Gauss1DMAP, 138
 - mappel::Gauss1DMLE, 167
 - mappel::Gauss1DModel, 195
 - mappel::Gauss2DMAP, 312
 - mappel::Gauss2DMLE, 343
 - mappel::Gauss2DModel, 374
 - mappel::Gauss2DsxyMAP, 502
- get_rng_generator
 - mappel::Gauss1DMAP, 138
 - mappel::Gauss1DMLE, 167
 - mappel::Gauss1DModel, 196
- mappel::Gauss1DsMAP, 224
- mappel::Gauss1DsMLE, 253
- mappel::Gauss1DsModel, 282
- mappel::Gauss2DMAP, 312
- mappel::Gauss2DMLE, 344
- mappel::Gauss2DModel, 374
- mappel::Gauss2DsMAP, 406
- mappel::Gauss2DsMLE, 440
- mappel::Gauss2DsModel, 472
- mappel::Gauss2DsxyMAP, 502
- mappel::Gauss2DsxyModel, 530
- mappel::MAPEstimator, 596
- mappel::MCMCAdaptor1Ds, 641
- mappel::MCMCAdaptor1D, 622
- mappel::MCMCAdaptor2Ds, 680
- mappel::MCMCAdaptor2D, 661
- mappel::MLEstimator, 710
- mappel::PointEmitterModel, 774
- get_rng_manager
 - mappel::Gauss1DMAP, 138
 - mappel::Gauss1DMLE, 167
 - mappel::Gauss1DModel, 196
 - mappel::Gauss1DsMAP, 224
 - mappel::Gauss1DsMLE, 253
 - mappel::Gauss1DsModel, 282
 - mappel::Gauss2DMAP, 313
 - mappel::Gauss2DMLE, 344
 - mappel::Gauss2DModel, 374
 - mappel::Gauss2DsMAP, 406
 - mappel::Gauss2DsMLE, 440
 - mappel::Gauss2DsModel, 472
 - mappel::Gauss2DsxyMAP, 502
 - mappel::Gauss2DsxyModel, 530
 - mappel::MAPEstimator, 597
 - mappel::MCMCAdaptor1Ds, 641
 - mappel::MCMCAdaptor1D, 622
 - mappel::MCMCAdaptor2Ds, 680
 - mappel::MCMCAdaptor2D, 661
 - mappel::MLEstimator, 710
 - mappel::PointEmitterModel, 774
- get_sequence_len
 - mappel::estimator::IterativeMaximizer::Maximizer←
Data, 609
- get_size
 - mappel::Gauss1DMAP, 139
 - mappel::Gauss1DMLE, 168
 - mappel::Gauss1DModel, 196
 - mappel::Gauss1DsMAP, 224
 - mappel::Gauss1DsMLE, 254
 - mappel::Gauss1DsModel, 283
 - mappel::Gauss2DMAP, 313
 - mappel::Gauss2DMLE, 344
 - mappel::Gauss2DModel, 375
 - mappel::Gauss2DsMAP, 406, 407

- mappel::Gauss2DsMLE, 440
- mappel::Gauss2DsModel, 473
- mappel::Gauss2DsxyMAP, 503
- mappel::Gauss2DsxyModel, 530
- mappel::ImageFormat1DBase, 560, 561
- mappel::ImageFormat2DBase, 567
- mappel::PoissonNoise1DObjective, 790, 791
- mappel::PoissonNoise2DObjective, 798
- get_size_image_stack
 - mappel::Gauss1DMAP, 139
 - mappel::Gauss1DMLE, 168
 - mappel::Gauss1DModel, 196
 - mappel::Gauss1DsMAP, 225
 - mappel::Gauss1DsMLE, 254
 - mappel::Gauss1DsModel, 283
 - mappel::Gauss2DMAP, 313
 - mappel::Gauss2DMLE, 344
 - mappel::Gauss2DModel, 375
 - mappel::Gauss2DsMAP, 407
 - mappel::Gauss2DsMLE, 440
 - mappel::Gauss2DsModel, 473
 - mappel::Gauss2DsxyMAP, 503
 - mappel::Gauss2DsxyModel, 531
 - mappel::ImageFormat1DBase, 561
 - mappel::ImageFormat2DBase, 567
 - mappel::PoissonNoise1DObjective, 791
 - mappel::PoissonNoise2DObjective, 798
- get_stats
 - mappel::Gauss1DMAP, 139
 - mappel::Gauss1DMLE, 168
 - mappel::Gauss1DModel, 196
 - mappel::Gauss1DsMAP, 225
 - mappel::Gauss1DsMLE, 254
 - mappel::Gauss1DsModel, 283
 - mappel::Gauss2DMAP, 313
 - mappel::Gauss2DMLE, 344
 - mappel::Gauss2DModel, 375
 - mappel::Gauss2DsMAP, 407
 - mappel::Gauss2DsMLE, 440
 - mappel::Gauss2DsModel, 473
 - mappel::Gauss2DsxyMAP, 503
 - mappel::Gauss2DsxyModel, 531
 - mappel::ImageFormat1DBase, 561
 - mappel::ImageFormat2DBase, 567
 - mappel::MAPEstimator, 597
 - mappel::MCMCAdaptor1Ds, 641
 - mappel::MCMCAdaptor1D, 622
 - mappel::MCMCAdaptor2Ds, 680
 - mappel::MCMCAdaptor2D, 661
 - mappel::MCMCAdaptorBase, 694
 - mappel::MLEstimator, 711
 - mappel::PointEmitterModel, 774
 - mappel::PoissonNoise1DObjective, 791
 - mappel::PoissonNoise2DObjective, 798
- mappel::estimator::CGaussHeuristicEstimator, 101
- mappel::estimator::CGaussMLE, 113
- mappel::estimator::Estimator, 125
- mappel::estimator::HeuristicEstimator, 554
- mappel::estimator::IterativeMaximizer, 581
- mappel::estimator::NewtonDiagonalMaximizer, 733
- mappel::estimator::NewtonMaximizer, 754
- mappel::estimator::QuasiNewtonMaximizer, 819
- mappel::estimator::SimulatedAnnealingMaximizer, 837
- mappel::estimator::ThreadedEstimator, 873
- mappel::estimator::TrustRegionMaximizer, 889
- get_theta_sequence
 - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 609
- get_theta_sequence_rllh
 - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 609
- get_total_backtracks
 - mappel::estimator::IterativeMaximizer, 581
 - mappel::estimator::NewtonDiagonalMaximizer, 733
 - mappel::estimator::NewtonMaximizer, 754
 - mappel::estimator::QuasiNewtonMaximizer, 819
 - mappel::estimator::TrustRegionMaximizer, 889
- get_total_der_evals
 - mappel::estimator::IterativeMaximizer, 582
 - mappel::estimator::NewtonDiagonalMaximizer, 734
 - mappel::estimator::NewtonMaximizer, 755
 - mappel::estimator::QuasiNewtonMaximizer, 820
 - mappel::estimator::TrustRegionMaximizer, 890
- get_total_fun_evals
 - mappel::estimator::IterativeMaximizer, 582
 - mappel::estimator::NewtonDiagonalMaximizer, 734
 - mappel::estimator::NewtonMaximizer, 755
 - mappel::estimator::QuasiNewtonMaximizer, 820
 - mappel::estimator::TrustRegionMaximizer, 890
- get_total_iterations
 - mappel::estimator::IterativeMaximizer, 582
 - mappel::estimator::NewtonDiagonalMaximizer, 734
 - mappel::estimator::NewtonMaximizer, 755
 - mappel::estimator::QuasiNewtonMaximizer, 820
 - mappel::estimator::TrustRegionMaximizer, 890
- get_ubound
 - mappel::Gauss1DMAP, 139
 - mappel::Gauss1DMLE, 168
 - mappel::Gauss1DModel, 196
 - mappel::Gauss1DsMAP, 225
 - mappel::Gauss1DsMLE, 254
 - mappel::Gauss1DsModel, 283
 - mappel::Gauss2DMAP, 313
 - mappel::Gauss2DMLE, 344
 - mappel::Gauss2DModel, 375
 - mappel::Gauss2DsMAP, 407
 - mappel::Gauss2DsMLE, 441

- mappel::Gauss2DsModel, [473](#)
- mappel::Gauss2DsxyMAP, [503](#)
- mappel::Gauss2DsxyModel, [531](#)
- mappel::MAPEstimator, [597](#)
- mappel::MCMCAdaptor1Ds, [641](#)
- mappel::MCMCAdaptor1D, [622](#)
- mappel::MCMCAdaptor2Ds, [680](#)
- mappel::MCMCAdaptor2D, [661](#)
- mappel::MLEstimator, [711](#)
- mappel::PointEmitterModel, [775](#)
- global_default_mcmc_sigma_scale
 - mappel::Gauss1DMAP, [152](#)
 - mappel::Gauss1DMLE, [181](#)
 - mappel::Gauss1DModel, [209](#)
 - mappel::Gauss1DsMAP, [239](#)
 - mappel::Gauss1DsMLE, [268](#)
 - mappel::Gauss1DsModel, [297](#)
 - mappel::Gauss2DMAP, [327](#)
 - mappel::Gauss2DMLE, [358](#)
 - mappel::Gauss2DModel, [389](#)
 - mappel::Gauss2DsMAP, [422](#)
 - mappel::Gauss2DsMLE, [455](#)
 - mappel::Gauss2DsModel, [487](#)
 - mappel::Gauss2DsxyMAP, [516](#)
 - mappel::MCMCAdaptor1Ds, [651](#)
 - mappel::MCMCAdaptor1D, [632](#)
 - mappel::MCMCAdaptor2Ds, [690](#)
 - mappel::MCMCAdaptor2D, [671](#)
 - mappel::MCMCAdaptorBase, [695](#)
- global_max_mcmc_sigma_scale
 - mappel::Gauss1DMAP, [152](#)
 - mappel::Gauss1DMLE, [181](#)
 - mappel::Gauss1DModel, [209](#)
 - mappel::Gauss1DsMAP, [239](#)
 - mappel::Gauss1DsMLE, [268](#)
 - mappel::Gauss1DsModel, [297](#)
 - mappel::Gauss2DMAP, [327](#)
 - mappel::Gauss2DMLE, [358](#)
 - mappel::Gauss2DModel, [389](#)
 - mappel::Gauss2DsMAP, [422](#)
 - mappel::Gauss2DsMLE, [455](#)
 - mappel::Gauss2DsModel, [488](#)
 - mappel::Gauss2DsxyMAP, [516](#)
 - mappel::Gauss2DsxyModel, [541](#)
 - mappel::ImageFormat1DBase, [562](#)
 - mappel::ImageFormat2DBase, [568](#)
 - mappel::PoissonNoise1DObjective, [792](#)
 - mappel::PoissonNoise2DObjective, [799](#)
- global_min_psf_sigma
 - mappel::Gauss1DMAP, [153](#)
 - mappel::Gauss1DMLE, [182](#)
 - mappel::Gauss1DModel, [210](#)
 - mappel::Gauss1DsMAP, [239](#)
 - mappel::Gauss1DsMLE, [269](#)
 - mappel::Gauss1DsModel, [297](#)
 - mappel::Gauss2DMAP, [328](#)
 - mappel::Gauss2DMLE, [359](#)
 - mappel::Gauss2DModel, [389](#)
 - mappel::Gauss2DsMAP, [423](#)
 - mappel::Gauss2DsMLE, [456](#)
 - mappel::Gauss2DsModel, [488](#)
 - mappel::Gauss2DsxyMAP, [516](#)
 - mappel::Gauss2DsxyModel, [541](#)
 - mappel::MAPEstimator, [605](#)
 - mappel::MCMCAdaptor1Ds, [652](#)
 - mappel::MCMCAdaptor1D, [632](#)
 - mappel::MCMCAdaptor2Ds, [691](#)
- mappel::Gauss1DsModel, [297](#)
- mappel::Gauss2DMAP, [328](#)
- mappel::Gauss2DMLE, [359](#)
- mappel::Gauss2DModel, [389](#)
- mappel::Gauss2DsMAP, [422](#)
- mappel::Gauss2DsMLE, [456](#)
- mappel::Gauss2DsModel, [488](#)
- mappel::Gauss2DsxyMAP, [516](#)
- mappel::Gauss2DsxyModel, [540](#)
- mappel::MAPEstimator, [605](#)
- mappel::MCMCAdaptor1Ds, [651](#)
- mappel::MCMCAdaptor1D, [632](#)
- mappel::MCMCAdaptor2Ds, [691](#)
- mappel::MCMCAdaptor2D, [671](#)
- mappel::MLEstimator, [719](#)
- mappel::PointEmitterModel, [782](#)

- mappel::MCMCAdaptor2D, 671
- mappel::MLEstimator, 719
- mappel::PointEmitterModel, 782
- global_min_size
 - mappel::Gauss1DModel, 153
 - mappel::Gauss1DModel, 182
 - mappel::Gauss1DModel, 210
 - mappel::Gauss1DsMAP, 240
 - mappel::Gauss1DsMLE, 269
 - mappel::Gauss1DsModel, 297
 - mappel::Gauss2DModel, 328
 - mappel::Gauss2DModel, 359
 - mappel::Gauss2DModel, 389
 - mappel::Gauss2DsMAP, 423
 - mappel::Gauss2DsMLE, 456
 - mappel::Gauss2DsModel, 488
 - mappel::Gauss2DsxyMAP, 517
 - mappel::Gauss2DsxyModel, 541
 - mappel::ImageFormat1DBase, 562
 - mappel::ImageFormat2DBase, 568
 - mappel::PoissonNoise1DObjective, 792
 - mappel::PoissonNoise2DObjective, 799
- grad
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, 613
 - mappel::methods::likelihood, 68
 - mappel::methods::objective, 73, 74
- grad2
 - mappel::methods::likelihood, 68, 69
 - mappel::methods::objective, 74
- grad_components
 - mappel::methods::likelihood::debug, 71
 - mappel::methods::objective::debug, 78
- grad_stack
 - mappel::methods::objective::openmp, 80
- GradRatio
 - mappel::estimator, 50
- Gx
 - mappel::Gauss1DModel::Stencil, 862
 - mappel::Gauss1DsModel::Stencil, 848
 - mappel::Gauss2DModel::Stencil, 853
 - mappel::Gauss2DsModel::Stencil, 858
 - mappel::Gauss2DsxyModel::Stencil, 844
- Gy
 - mappel::Gauss2DModel::Stencil, 853
 - mappel::Gauss2DsModel::Stencil, 859
 - mappel::Gauss2DsxyModel::Stencil, 844
- has_fixed_parameters
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, 609
- has_hyperparam
 - mappel::Gauss1DModel, 139
 - mappel::Gauss1DModel, 168
- mappel::Gauss1DModel, 197
- mappel::Gauss1DsMAP, 225
- mappel::Gauss1DsMLE, 254
- mappel::Gauss1DsModel, 283
- mappel::Gauss2DModel, 313
- mappel::Gauss2DModel, 345
- mappel::Gauss2DModel, 375
- mappel::Gauss2DsMAP, 407
- mappel::Gauss2DsMLE, 441
- mappel::Gauss2DsModel, 473
- mappel::Gauss2DsxyMAP, 503
- mappel::Gauss2DsxyModel, 531
- mappel::MAPEstimator, 597
- mappel::MCMCAdaptor1Ds, 642
- mappel::MCMCAdaptor1D, 623
- mappel::MCMCAdaptor2Ds, 681
- mappel::MCMCAdaptor2D, 661
- mappel::MLEstimator, 711
- mappel::PointEmitterModel, 775
- has_theta_sequence
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, 609
- hessian
 - mappel::methods::likelihood, 69
 - mappel::methods::objective, 74, 75
- hessian_components
 - mappel::methods::likelihood::debug, 71
 - mappel::methods::objective::debug, 78
- hessian_stack
 - mappel::methods::objective::openmp, 80
- HeuristicEstimator
 - mappel::estimator::HeuristicEstimator, 546
- I
 - mappel::Gauss1DModel::Stencil, 861
 - mappel::Gauss1DsModel::Stencil, 847
 - mappel::Gauss2DModel::Stencil, 851
 - mappel::Gauss2DsModel::Stencil, 856
 - mappel::Gauss2DsxyModel::Stencil, 842
- icontains
 - mappel, 36
- IdxMatT
 - mappel, 27
- IdxVecT
 - mappel, 27
- IdxT
 - mappel, 27
- im
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, 613
- ImageCoordT
 - mappel, 27
 - mappel::Gauss1DModel, 132
 - mappel::Gauss1DModel, 161

- mappel::Gauss1DModel, 190
- mappel::Gauss1DsMAP, 217
- mappel::Gauss1DsMLE, 247
- mappel::Gauss1DsModel, 276
- mappel::Gauss2DMAP, 305
- mappel::Gauss2DMLE, 337
- mappel::Gauss2DModel, 368
- mappel::Gauss2DsMAP, 398
- mappel::Gauss2DsMLE, 432
- mappel::Gauss2DsModel, 465
- mappel::Gauss2DsxyMAP, 496
- mappel::Gauss2DsxyModel, 524
- mappel::ImageFormat1DBase, 558
- mappel::ImageFormat2DBase, 564
- mappel::PoissonNoise1DObjective, 788
- mappel::PoissonNoise2DObjective, 795
- ImageFormat1DBase
 - mappel::ImageFormat1DBase, 560
- ImageFormat1DBase.cpp, 924
- ImageFormat1DBase.h, 925
- ImageFormat2DBase
 - mappel::ImageFormat2DBase, 566
- ImageFormat2DBase.cpp, 926
- ImageFormat2DBase.h, 926
- ImagePixelT
 - mappel, 27
 - mappel::Gauss1DMAP, 132
 - mappel::Gauss1DMLE, 161
 - mappel::Gauss1DModel, 190
 - mappel::Gauss1DsMAP, 217
 - mappel::Gauss1DsMLE, 247
 - mappel::Gauss1DsModel, 276
 - mappel::Gauss2DMAP, 305
 - mappel::Gauss2DMLE, 337
 - mappel::Gauss2DModel, 368
 - mappel::Gauss2DsMAP, 398
 - mappel::Gauss2DsMLE, 432
 - mappel::Gauss2DsModel, 465
 - mappel::Gauss2DsxyMAP, 496
 - mappel::Gauss2DsxyModel, 524
 - mappel::ImageFormat1DBase, 558
 - mappel::ImageFormat2DBase, 564
 - mappel::PoissonNoise1DObjective, 788
 - mappel::PoissonNoise2DObjective, 795
- ImageShapeT
 - mappel::Gauss1DMAP, 132
 - mappel::Gauss1DMLE, 161
 - mappel::Gauss1DModel, 190
 - mappel::Gauss1DsMAP, 218
 - mappel::Gauss1DsMLE, 247
 - mappel::Gauss1DsModel, 277
 - mappel::Gauss2DMAP, 306
 - mappel::Gauss2DMLE, 337
 - mappel::Gauss2DModel, 368
 - mappel::Gauss2DsMAP, 399
 - mappel::Gauss2DsMLE, 432
 - mappel::Gauss2DsModel, 466
 - mappel::Gauss2DsxyMAP, 497
 - mappel::Gauss2DsxyModel, 525
 - mappel::ImageFormat1DBase, 559
 - mappel::ImageFormat2DBase, 565
 - mappel::PoissonNoise1DObjective, 788
 - mappel::PoissonNoise2DObjective, 796
- ImageSizeShapeT
 - mappel::Gauss1DMAP, 132
 - mappel::Gauss1DMLE, 161
 - mappel::Gauss1DModel, 190
 - mappel::Gauss1DsMAP, 218
 - mappel::Gauss1DsMLE, 247
 - mappel::Gauss1DsModel, 277
 - mappel::Gauss2DMAP, 306
 - mappel::Gauss2DMLE, 337
 - mappel::Gauss2DModel, 368
 - mappel::Gauss2DsMAP, 399
 - mappel::Gauss2DsMLE, 432
 - mappel::Gauss2DsModel, 466
 - mappel::Gauss2DsxyMAP, 496
 - mappel::Gauss2DsxyModel, 524
 - mappel::ImageFormat1DBase, 559
 - mappel::ImageFormat2DBase, 565
 - mappel::PoissonNoise1DObjective, 788
 - mappel::PoissonNoise2DObjective, 795
- ImageSizeVecShapeT
 - mappel::Gauss1DMAP, 132
 - mappel::Gauss1DMLE, 161
 - mappel::Gauss1DModel, 190
 - mappel::Gauss1DsMAP, 218
 - mappel::Gauss1DsMLE, 247
 - mappel::Gauss1DsModel, 277
 - mappel::Gauss2DMAP, 306
 - mappel::Gauss2DMLE, 337
 - mappel::Gauss2DModel, 368
 - mappel::Gauss2DsMAP, 399
 - mappel::Gauss2DsMLE, 432
 - mappel::Gauss2DsModel, 466
 - mappel::Gauss2DsxyMAP, 497
 - mappel::Gauss2DsxyModel, 525
 - mappel::ImageFormat1DBase, 559
 - mappel::ImageFormat2DBase, 565
 - mappel::PoissonNoise1DObjective, 788
 - mappel::PoissonNoise2DObjective, 796
- ImageSizeVecT
 - mappel::Gauss1DMAP, 132
 - mappel::Gauss1DMLE, 161
 - mappel::Gauss1DModel, 190
 - mappel::Gauss1DsMAP, 218
 - mappel::Gauss1DsMLE, 247
 - mappel::Gauss1DsModel, 277

- mappel::Gauss2DModel, 306
- mappel::Gauss2DsMAP, 399
- mappel::Gauss2DsMLE, 433
- mappel::Gauss2DsModel, 466
- mappel::Gauss2DsxyMAP, 497
- mappel::Gauss2DsxyModel, 525
- mappel::ImageFormat1DBase, 559
- mappel::ImageFormat2DBase, 565
- mappel::PoissonNoise1DObjective, 788
- mappel::PoissonNoise2DObjective, 796
- ImageSizeT
 - mappel::Gauss1DModel, 190
 - mappel::Gauss1DsMAP, 218
 - mappel::Gauss1DsMLE, 247
 - mappel::Gauss1DsModel, 277
 - mappel::Gauss2DModel, 368
 - mappel::Gauss2DsMAP, 399
 - mappel::Gauss2DsMLE, 432
 - mappel::Gauss2DsModel, 466
 - mappel::Gauss2DsxyMAP, 497
 - mappel::Gauss2DsxyModel, 525
 - mappel::ImageFormat1DBase, 559
 - mappel::ImageFormat2DBase, 565
 - mappel::PoissonNoise1DObjective, 788
 - mappel::PoissonNoise2DObjective, 795
- ImageStackShapeT
 - mappel::Gauss1DModel, 190
 - mappel::Gauss1DsMAP, 218
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 277
 - mappel::Gauss2DModel, 369
 - mappel::Gauss2DsMAP, 399
 - mappel::Gauss2DsMLE, 433
 - mappel::Gauss2DsModel, 466
 - mappel::Gauss2DsxyMAP, 497
 - mappel::Gauss2DsxyModel, 525
 - mappel::ImageFormat1DBase, 559
 - mappel::ImageFormat2DBase, 565
 - mappel::PoissonNoise1DObjective, 789
 - mappel::PoissonNoise2DObjective, 796
- ImageStackT
 - mappel, 27
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DsMAP, 218
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 277
 - mappel::Gauss2DModel, 369
 - mappel::Gauss2DsMAP, 399
 - mappel::Gauss2DsMLE, 433
 - mappel::Gauss2DsModel, 466
 - mappel::Gauss2DsxyMAP, 497
 - mappel::Gauss2DsxyModel, 525
 - mappel::ImageFormat1DBase, 559
 - mappel::ImageFormat2DBase, 565
 - mappel::PoissonNoise1DObjective, 789
 - mappel::PoissonNoise2DObjective, 796
- ImageT
 - mappel, 27
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DsMAP, 219
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 277
 - mappel::Gauss2DModel, 369
 - mappel::Gauss2DsMAP, 400
 - mappel::Gauss2DsMLE, 433
 - mappel::Gauss2DsModel, 466
 - mappel::Gauss2DsxyMAP, 497
 - mappel::Gauss2DsxyModel, 525
 - mappel::ImageFormat1DBase, 559
 - mappel::ImageFormat2DBase, 565
 - mappel::PoissonNoise1DObjective, 789
 - mappel::PoissonNoise2DObjective, 796
- initial_theta_estimate
 - mappel::Gauss1DModel, 197
 - mappel::Gauss1DsMAP, 225
 - mappel::Gauss1DsMLE, 254, 255
 - mappel::Gauss1DsModel, 283, 284
 - mappel::Gauss2DModel, 375, 376
 - mappel::Gauss2DsMAP, 407, 408
 - mappel::Gauss2DsMLE, 441
 - mappel::Gauss2DsModel, 473, 474
 - mappel::Gauss2DsxyMAP, 503, 504
 - mappel::Gauss2DsxyModel, 531
- initialize_arrays
 - mappel::estimator::ProfileBoundsData, 801
 - mappel::estimator::ProfileBoundsDataStack, 805

- mappel::mcmc::MCMCData, 696
 - mappel::mcmc::MCMCDataStack, 699
 - mappel::mcmc::MCMCDebugData, 701
- Install.md, 927
- is_negative_definite
 - mappel, 36
- is_positive_definite
 - mappel, 37
- is_symmetric
 - mappel, 37
- istarts_with
 - mappel, 37
- IterativeMaximizer
 - mappel::estimator::IterativeMaximizer, 572
- lambda_term_color
 - mappel, 38
- last_backtrack_idx
 - mappel::estimator::IterativeMaximizer, 586
 - mappel::estimator::NewtonDiagonalMaximizer, 739
 - mappel::estimator::NewtonMaximizer, 760
 - mappel::estimator::QuasiNewtonMaximizer, 825
 - mappel::estimator::TrustRegionMaximizer, 895
- lbound
 - mappel::Gauss1DMap, 153
 - mappel::Gauss1DMLE, 182
 - mappel::Gauss1DModel, 210
 - mappel::Gauss1DsMap, 240
 - mappel::Gauss1DsMLE, 269
 - mappel::Gauss1DsModel, 298
 - mappel::Gauss2DMap, 328
 - mappel::Gauss2DMLE, 359
 - mappel::Gauss2DModel, 390
 - mappel::Gauss2DsMap, 423
 - mappel::Gauss2DsMLE, 456
 - mappel::Gauss2DsModel, 488
 - mappel::Gauss2DsxyMap, 517
 - mappel::Gauss2DsxyModel, 541
 - mappel::MAPEstimator, 605
 - mappel::MCMCAdaptor1Ds, 652
 - mappel::MCMCAdaptor1D, 632
 - mappel::MCMCAdaptor2Ds, 691
 - mappel::MCMCAdaptor2D, 671
 - mappel::MLEstimator, 719
 - mappel::PointEmitterModel, 783
- likelihood_objective
 - mappel::methods, 62
- llh
 - mappel::methods::likelihood, 69
 - mappel::methods::objective, 75
- llh_components
 - mappel::methods::likelihood::debug, 71
 - mappel::methods::objective::debug, 78, 79
- llh_stack
 - mappel::methods::objective::openmp, 81, 82
- local_maximize
 - mappel::estimator::IterativeMaximizer, 582
 - mappel::estimator::NewtonDiagonalMaximizer, 734
 - mappel::estimator::NewtonMaximizer, 755
 - mappel::estimator::QuasiNewtonMaximizer, 820
 - mappel::estimator::TrustRegionMaximizer, 890
- local_profile_maximize
 - mappel::estimator::IterativeMaximizer, 582
 - mappel::estimator::NewtonDiagonalMaximizer, 734
 - mappel::estimator::NewtonMaximizer, 755
 - mappel::estimator::QuasiNewtonMaximizer, 820
 - mappel::estimator::TrustRegionMaximizer, 890
- log_likelihood
 - PoissonGaussianNoise2DObjective.h, 953
- 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, 590
- MAPEstimator
 - mappel::MAPEstimator, 593, 594
- MAPEstimator.h, 927
- MCMCAdaptor1D.cpp, 931
- MCMCAdaptor1D.h, 931
- MCMCAdaptor1Ds
 - mappel::MCMCAdaptor1Ds, 638
- MCMCAdaptor1Ds.cpp, 932
- MCMCAdaptor1Ds.h, 933
- MCMCAdaptor1D
 - mappel::MCMCAdaptor1D, 619
- MCMCAdaptor2D.cpp, 933
- MCMCAdaptor2D.h, 934
- MCMCAdaptor2Ds
 - mappel::MCMCAdaptor2Ds, 677
- MCMCAdaptor2Ds.cpp, 934
- MCMCAdaptor2Ds.h, 935
- MCMCAdaptor2D
 - mappel::MCMCAdaptor2D, 657
- MCMCAdaptorBase
 - mappel::MCMCAdaptorBase, 694
- MCMCAdaptorBase.cpp, 935
- MCMCAdaptorBase.h, 936
- MLEstimator
 - mappel::MLEstimator, 707
- MLEstimator.h, 936

- 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, [169](#)
 - mappel::Gauss1DModel, [197](#)
 - mappel::Gauss1DsMAP, [225](#)
 - mappel::Gauss1DsMLE, [255](#)
 - mappel::Gauss1DsModel, [284](#)
 - mappel::Gauss2DMap, [314](#)
 - mappel::Gauss2DMLE, [345](#)
 - mappel::Gauss2DModel, [376](#)
 - mappel::Gauss2DsMAP, [408](#)
 - mappel::Gauss2DsMLE, [441](#)
 - mappel::Gauss2DsModel, [474](#)
 - mappel::Gauss2DsxyMAP, [504](#)
 - mappel::Gauss2DsxyModel, [532](#)
- make_default_prior_beta_position
 - mappel::Gauss1DMap, [140](#)
 - mappel::Gauss1DMLE, [169](#)
 - mappel::Gauss1DModel, [197](#)
 - mappel::Gauss1DsMAP, [226](#)
 - mappel::Gauss1DsMLE, [255](#)
 - mappel::Gauss1DsModel, [284](#)
 - mappel::Gauss2DMap, [314](#)
 - mappel::Gauss2DMLE, [346](#)
 - mappel::Gauss2DModel, [376](#)
 - mappel::Gauss2DsMAP, [408](#)
 - mappel::Gauss2DsMLE, [442](#)
 - mappel::Gauss2DsModel, [474](#)
 - mappel::Gauss2DsxyMAP, [504](#)
- make_default_prior_normal_position
 - mappel::Gauss1DMap, [140](#)
 - mappel::Gauss1DMLE, [169](#)
 - mappel::Gauss1DModel, [197](#)
 - mappel::Gauss1DsMAP, [226](#)
 - mappel::Gauss1DsMLE, [255](#)
 - mappel::Gauss1DsModel, [284](#)
 - mappel::Gauss2DMap, [314](#)
 - mappel::Gauss2DMLE, [346](#)
 - mappel::Gauss2DModel, [376](#)
 - mappel::Gauss2DsMAP, [408](#)
- mappel::Gauss2DsMLE, [442](#)
- mappel::Gauss2DsModel, [474](#)
- mappel::Gauss2DsxyMAP, [504](#)
- mappel::Gauss2DsMLE, [442](#)
- mappel::Gauss2DsModel, [474](#)
- mappel::Gauss2DsxyMAP, [504](#)
- make_estimator
 - mappel::methods, [63](#)
 - PoissonGaussianNoise2DObjective.h, [953](#)
- make_gaussian_stencil
 - mappel, [39](#)
- make_image
 - mappel::Gauss1DMap, [140](#)
 - mappel::Gauss1DMLE, [169](#)
 - mappel::Gauss1DModel, [198](#)
 - mappel::Gauss1DsMAP, [226](#)
 - mappel::Gauss1DsMLE, [255](#)
 - mappel::Gauss1DsModel, [284](#)
 - mappel::Gauss2DMap, [315](#)
 - mappel::Gauss2DMLE, [346](#)
 - mappel::Gauss2DModel, [376](#)
 - mappel::Gauss2DsMAP, [408](#)
 - mappel::Gauss2DsMLE, [442](#)
 - mappel::Gauss2DsModel, [475](#)
 - mappel::Gauss2DsxyMAP, [504](#)
 - mappel::Gauss2DsxyModel, [532](#)
 - mappel::ImageFormat1DBase, [561](#)
 - mappel::ImageFormat2DBase, [567](#)
 - mappel::PoissonNoise1DObjective, [791](#)
 - mappel::PoissonNoise2DObjective, [798](#)
- make_image_stack
 - mappel::Gauss1DMap, [140](#)
 - mappel::Gauss1DMLE, [169](#)
 - mappel::Gauss1DModel, [198](#)
 - mappel::Gauss1DsMAP, [226](#)
 - mappel::Gauss1DsMLE, [256](#)
 - mappel::Gauss1DsModel, [284](#)
 - mappel::Gauss2DMap, [315](#)
 - mappel::Gauss2DMLE, [346](#)
 - mappel::Gauss2DModel, [377](#)
 - mappel::Gauss2DsMAP, [409](#)
 - mappel::Gauss2DsMLE, [442](#)
 - mappel::Gauss2DsModel, [475](#)
 - mappel::Gauss2DsxyMAP, [504](#)
 - mappel::Gauss2DsxyModel, [532](#)
 - mappel::ImageFormat1DBase, [561](#)
 - mappel::ImageFormat2DBase, [567](#)
 - mappel::PoissonNoise1DObjective, [791](#)
 - mappel::PoissonNoise2DObjective, [798](#)
- make_internal_1Dsum_estimator
 - mappel::Gauss2DMap, [315](#)
 - mappel::Gauss2DMLE, [346](#)
 - mappel::Gauss2DModel, [377](#)
 - mappel::Gauss2DsMAP, [409](#)
 - mappel::Gauss2DsMLE, [442](#)
 - mappel::Gauss2DsModel, [475](#)
- make_param

- mappel::Gauss1DMAP, 141
- mappel::Gauss1DMLE, 170
- mappel::Gauss1DModel, 198
- mappel::Gauss1DsMAP, 226
- mappel::Gauss1DsMLE, 256
- mappel::Gauss1DsModel, 285
- mappel::Gauss2DMAP, 315
- mappel::Gauss2DMLE, 346, 347
- mappel::Gauss2DModel, 377
- mappel::Gauss2DsMAP, 409
- mappel::Gauss2DsMLE, 443
- mappel::Gauss2DsModel, 475
- mappel::Gauss2DsxyMAP, 505
- mappel::Gauss2DsxyModel, 532
- mappel::MAPEstimator, 597
- mappel::MCMCAdaptor1Ds, 642
- mappel::MCMCAdaptor1D, 623
- mappel::MCMCAdaptor2Ds, 681
- mappel::MCMCAdaptor2D, 661
- mappel::MLEstimator, 711
- mappel::PointEmitterModel, 775
- make_param_mat
 - mappel::Gauss1DMAP, 141
 - mappel::Gauss1DMLE, 170
 - mappel::Gauss1DModel, 198
 - mappel::Gauss1DsMAP, 227
 - mappel::Gauss1DsMLE, 256
 - mappel::Gauss1DsModel, 285
 - mappel::Gauss2DMAP, 316
 - mappel::Gauss2DMLE, 347
 - mappel::Gauss2DModel, 377
 - mappel::Gauss2DsMAP, 409
 - mappel::Gauss2DsMLE, 443
 - mappel::Gauss2DsModel, 475, 476
 - mappel::Gauss2DsxyMAP, 505
 - mappel::Gauss2DsxyModel, 532
 - mappel::MAPEstimator, 598
 - mappel::MCMCAdaptor1Ds, 642
 - mappel::MCMCAdaptor1D, 623
 - mappel::MCMCAdaptor2Ds, 681
 - mappel::MCMCAdaptor2D, 662
 - mappel::MLEstimator, 711, 712
 - mappel::PointEmitterModel, 775
- make_param_mat_stack
 - mappel::Gauss1DMAP, 141
 - mappel::Gauss1DMLE, 170
 - mappel::Gauss1DModel, 198, 199
 - mappel::Gauss1DsMAP, 227
 - mappel::Gauss1DsMLE, 256
 - mappel::Gauss1DsModel, 285
 - mappel::Gauss2DMAP, 316
 - mappel::Gauss2DMLE, 347
 - mappel::Gauss2DModel, 378
 - mappel::Gauss2DsMAP, 410
- mappel::Gauss2DsMLE, 443
- mappel::Gauss2DsModel, 476
- mappel::Gauss2DsxyMAP, 505
- mappel::Gauss2DsxyModel, 532, 533
- mappel::MAPEstimator, 598
- mappel::MCMCAdaptor1Ds, 642, 643
- mappel::MCMCAdaptor1D, 623, 624
- mappel::MCMCAdaptor2Ds, 681, 682
- mappel::MCMCAdaptor2D, 662
- mappel::MLEstimator, 712
- mappel::PointEmitterModel, 776
- make_param_stack
 - mappel::Gauss1DMAP, 141, 142
 - mappel::Gauss1DMLE, 170, 171
 - mappel::Gauss1DModel, 199
 - mappel::Gauss1DsMAP, 227
 - mappel::Gauss1DsMLE, 257
 - mappel::Gauss1DsModel, 285, 286
 - mappel::Gauss2DMAP, 316
 - mappel::Gauss2DMLE, 347
 - mappel::Gauss2DModel, 378
 - mappel::Gauss2DsMAP, 410
 - mappel::Gauss2DsMLE, 444
 - mappel::Gauss2DsModel, 476
 - mappel::Gauss2DsxyMAP, 505, 506
 - mappel::Gauss2DsxyModel, 533
 - mappel::MAPEstimator, 598
 - mappel::MCMCAdaptor1Ds, 643
 - mappel::MCMCAdaptor1D, 624
 - mappel::MCMCAdaptor2Ds, 682
 - mappel::MCMCAdaptor2D, 662
 - mappel::MLEstimator, 712
 - mappel::PointEmitterModel, 776
- make_prior_beta_position
 - mappel::Gauss1DMAP, 142
 - mappel::Gauss1DMLE, 171
 - mappel::Gauss1DModel, 199
 - mappel::Gauss1DsMAP, 227
 - mappel::Gauss1DsMLE, 257
 - mappel::Gauss1DsModel, 286
 - mappel::Gauss2DMAP, 316
 - mappel::Gauss2DMLE, 348
 - mappel::Gauss2DModel, 378
 - mappel::Gauss2DsMAP, 410
 - mappel::Gauss2DsMLE, 444
 - mappel::Gauss2DsModel, 476
 - mappel::Gauss2DsxyMAP, 506
 - mappel::Gauss2DsxyModel, 533
- make_prior_component_intensity
 - mappel::Gauss1DMAP, 142
 - mappel::Gauss1DMLE, 171
 - mappel::Gauss1DModel, 199
 - mappel::Gauss1DsMAP, 228
 - mappel::Gauss1DsMLE, 257

- mappel::Gauss1DsModel, [286](#)
- mappel::Gauss2DMAP, [317](#)
- mappel::Gauss2DMLE, [348](#)
- mappel::Gauss2DModel, [378](#)
- mappel::Gauss2DsMAP, [410](#)
- mappel::Gauss2DsMLE, [444](#)
- mappel::Gauss2DsModel, [476](#)
- mappel::Gauss2DsxyMAP, [506](#)
- mappel::Gauss2DsxyModel, [533](#)
- mappel::MAPEstimator, [598](#)
- mappel::MCMCAdaptor1Ds, [643](#)
- mappel::MCMCAdaptor1D, [624](#)
- mappel::MCMCAdaptor2Ds, [682](#)
- mappel::MCMCAdaptor2D, [662](#)
- mappel::MLEstimator, [712](#)
- mappel::PointEmitterModel, [776](#)
- make_prior_component_position_beta
 - mappel::Gauss1DMAP, [142](#)
 - mappel::Gauss1DMLE, [171](#)
 - mappel::Gauss1DModel, [199](#)
 - mappel::Gauss1DsMAP, [228](#)
 - mappel::Gauss1DsMLE, [257](#)
 - mappel::Gauss1DsModel, [286](#)
 - mappel::Gauss2DMAP, [317](#)
 - mappel::Gauss2DMLE, [348](#)
 - mappel::Gauss2DModel, [379](#)
 - mappel::Gauss2DsMAP, [411](#)
 - mappel::Gauss2DsMLE, [444](#)
 - mappel::Gauss2DsModel, [477](#)
 - mappel::Gauss2DsxyMAP, [506](#)
 - mappel::Gauss2DsxyModel, [533](#)
 - mappel::MAPEstimator, [599](#)
 - mappel::MCMCAdaptor1Ds, [643](#)
 - mappel::MCMCAdaptor1D, [624](#)
 - mappel::MCMCAdaptor2Ds, [682](#)
 - mappel::MCMCAdaptor2D, [663](#)
 - mappel::MLEstimator, [713](#)
 - mappel::PointEmitterModel, [776](#)
- make_prior_component_position_normal
 - mappel::Gauss1DMAP, [142](#)
 - mappel::Gauss1DMLE, [171](#)
 - mappel::Gauss1DModel, [200](#)
 - mappel::Gauss1DsMAP, [228](#)
 - mappel::Gauss1DsMLE, [258](#)
 - mappel::Gauss1DsModel, [286](#)
 - mappel::Gauss2DMAP, [317](#)
 - mappel::Gauss2DMLE, [348](#)
 - mappel::Gauss2DModel, [379](#)
 - mappel::Gauss2DsMAP, [411](#)
 - mappel::Gauss2DsMLE, [445](#)
 - mappel::Gauss2DsModel, [477](#)
 - mappel::Gauss2DsxyMAP, [506](#)
 - mappel::Gauss2DsxyModel, [534](#)
 - mappel::MAPEstimator, [599](#)
- mappel::MCMCAdaptor1Ds, [644](#)
- mappel::MCMCAdaptor1D, [625](#)
- mappel::MCMCAdaptor2Ds, [683](#)
- mappel::MCMCAdaptor2D, [663](#)
- mappel::MLEstimator, [713](#)
- mappel::PointEmitterModel, [777](#)
- make_prior_component_sigma
 - mappel::Gauss1DMAP, [143](#)
 - mappel::Gauss1DMLE, [172](#)
 - mappel::Gauss1DModel, [200](#)
 - mappel::Gauss1DsMAP, [228](#)
 - mappel::Gauss1DsMLE, [258](#)
 - mappel::Gauss1DsModel, [287](#)
 - mappel::Gauss2DMAP, [317](#)
 - mappel::Gauss2DMLE, [348](#)
 - mappel::Gauss2DModel, [379](#)
 - mappel::Gauss2DsMAP, [411](#)
 - mappel::Gauss2DsMLE, [445](#)
 - mappel::Gauss2DsModel, [477](#)
 - mappel::Gauss2DsxyMAP, [507](#)
 - mappel::Gauss2DsxyModel, [534](#)
 - mappel::MAPEstimator, [599](#)
 - mappel::MCMCAdaptor1Ds, [644](#)
 - mappel::MCMCAdaptor1D, [625](#)
 - mappel::MCMCAdaptor2Ds, [683](#)
 - mappel::MCMCAdaptor2D, [663](#)
 - mappel::MLEstimator, [713](#)
 - mappel::PointEmitterModel, [777](#)
- make_prior_normal_position
 - mappel::Gauss1DMAP, [143](#)
 - mappel::Gauss1DMLE, [172](#)
 - mappel::Gauss1DModel, [200](#)
 - mappel::Gauss1DsMAP, [229](#)
 - mappel::Gauss1DsMLE, [258](#)
 - mappel::Gauss1DsModel, [287](#)
 - mappel::Gauss2DMAP, [318](#)
 - mappel::Gauss2DMLE, [349](#)
 - mappel::Gauss2DModel, [379](#)
 - mappel::Gauss2DsMAP, [411](#)
 - mappel::Gauss2DsMLE, [445](#)
 - mappel::Gauss2DsModel, [477](#)
 - mappel::Gauss2DsxyMAP, [507](#)
 - mappel::Gauss2DsxyModel, [534](#)
- make_stencil
 - mappel::Gauss1DMAP, [143](#)
 - mappel::Gauss1DMLE, [172](#)
 - mappel::Gauss1DModel, [200](#)
 - mappel::Gauss1DsMAP, [229](#)
 - mappel::Gauss1DsMLE, [258](#)
 - mappel::Gauss1DsModel, [287](#)
 - mappel::Gauss2DMAP, [318](#)
 - mappel::Gauss2DMLE, [349](#)
 - mappel::Gauss2DModel, [379](#)
 - mappel::Gauss2DsMAP, [412](#)

- mappel::Gauss2DsMLE, [445](#)
 - mappel::Gauss2DsModel, [477](#)
 - mappel::Gauss2DsxyMAP, [507](#)
 - mappel::Gauss2DsxyModel, [534](#)
- make_unique
 - mappel, [39](#)
- makeMLEData
 - mappel::estimator::MLEDebugData, [702](#)
- 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, [90](#)
 - ArrayShapeError, [91](#)
- mappel::ArraySizeError, [91](#)
 - ArraySizeError, [91](#)
- mappel::Gauss1DMAP, [127](#)
 - bound_theta, [134](#)
 - bounded_theta, [134](#)
 - bounded_theta_stack, [135](#)
 - bounds_epsilon, [149](#)
 - check_image_shape, [135](#)
 - check_param_shape, [135](#)
 - check_psf_sigma, [135](#), [136](#)
 - check_size, [136](#)
 - DefaultConfidenceLevel, [149](#)
 - DefaultEstimatorMethod, [150](#)
 - DefaultMCMCBurnin, [150](#)
 - DefaultMCMCNumSamples, [150](#)
 - DefaultMCMCThin, [150](#)
 - DefaultPriorBetaPos, [150](#)
 - DefaultPriorIntensityKappa, [150](#)
 - DefaultPriorMaxI, [150](#)
 - DefaultPriorMeanI, [150](#)
 - DefaultPriorPSFSigmaAlpha, [151](#)
 - DefaultPriorPixelMeanBG, [151](#)
 - DefaultPriorSigmaPos, [151](#)
 - DefaultPriorType, [151](#)
 - DefaultProfileBoundsEstimatorMethod, [151](#)
 - DefaultSeperableInitEstimator, [151](#)
 - estimator_names, [152](#)
 - eta_bg, [152](#)
 - eta_l, [152](#)
 - eta_x, [152](#)
 - Gauss1DMAP, [134](#)
 - get_hyperparam_index, [136](#)
 - get_hyperparam_names, [136](#)
 - get_hyperparam_value, [136](#)
 - get_hyperparams, [136](#)
 - get_image_from_stack, [137](#)
 - get_lbound, [137](#)
 - get_mcmc_num_phases, [137](#)
 - get_mcmc_sigma_scale, [137](#)
 - get_num_hyperparams, [137](#)
 - get_num_params, [137](#)
 - get_num_pixels, [137](#)
 - get_param_names, [138](#)
 - get_prior, [138](#)
 - get_psf_sigma, [138](#)
 - get_rng_generator, [138](#)
 - get_rng_manager, [138](#)
 - get_size, [139](#)
 - get_size_image_stack, [139](#)
 - get_stats, [139](#)
 - get_ubound, [139](#)
 - global_default_mcmc_sigma_scale, [152](#)
 - global_max_mcmc_sigma_scale, [152](#)
 - global_max_psf_sigma, [153](#)
 - global_max_size, [153](#)
 - global_min_psf_sigma, [153](#)
 - global_min_size, [153](#)
 - has_hyperparam, [139](#)
 - ImageCoordT, [132](#)
 - ImagePixelT, [132](#)
 - ImageShapeT, [132](#)
 - ImageSizeShapeT, [132](#)
 - ImageSizeVecShapeT, [132](#)
 - ImageSizeVecT, [132](#)
 - ImageSizeT, [132](#)
 - ImageStackShapeT, [133](#)
 - ImageStackT, [133](#)
 - ImageT, [133](#)
 - initial_theta_estimate, [139](#), [140](#)
 - lbound, [153](#)
 - make_default_prior, [140](#)

- make_default_prior_beta_position, 140
- make_default_prior_normal_position, 140
- make_image, 140
- make_image_stack, 140
- make_param, 141
- make_param_mat, 141
- make_param_mat_stack, 141
- make_param_stack, 141, 142
- make_prior_beta_position, 142
- make_prior_component_intensity, 142
- make_prior_component_position_beta, 142
- make_prior_component_position_normal, 142
- make_prior_component_sigma, 143
- make_prior_normal_position, 143
- make_stencil, 143
- ModelDataStackT, 133
- ModelDataT, 133
- name, 153
- num_dim, 154
- num_hyperparams, 154
- num_params, 154
- num_phases, 154
- operator=, 144
- ParamVecT, 133
- ParamT, 133
- pixel_grad, 144
- pixel_grad2, 144
- pixel_hess, 144
- pixel_hess_update, 144
- pixel_model_value, 145
- prior, 154
- prior_types, 155
- psf_sigma, 155
- reflected_theta, 145
- reflected_theta_stack, 145
- rename_hyperparam, 145
- sample_mcmc_candidate, 145
- sample_prior, 146
- set_background_mcmc_sampling, 146
- set_bounds, 146
- set_hyperparam_names, 146
- set_hyperparam_value, 146
- set_hyperparams, 147
- set_image_in_stack, 147
- set_intensity_mcmc_sampling, 147
- set_lbound, 147
- set_mcmc_num_phases, 147
- set_mcmc_sigma_scale, 147
- set_param_names, 148
- set_prior, 148
- set_psf_sigma, 148
- set_rng_seed, 148
- set_size, 148, 149
- set_ubound, 149
- sigma_scale, 155
- size, 155
- StencilVecT, 133
- theta_in_bounds, 149
- theta_stack_in_bounds, 149
- ubound, 156
- mappel::Gauss1DMLE, 156
 - bound_theta, 163
 - bounded_theta, 163
 - bounded_theta_stack, 164
 - bounds_epsilon, 178
 - check_image_shape, 164
 - check_param_shape, 164
 - check_psf_sigma, 164, 165
 - check_size, 165
 - DefaultConfidenceLevel, 178
 - DefaultEstimatorMethod, 179
 - DefaultMCMCBurnin, 179
 - DefaultMCMCNumSamples, 179
 - DefaultMCMCThin, 179
 - DefaultPriorBetaPos, 179
 - DefaultPriorIntensityKappa, 179
 - DefaultPriorMaxI, 179
 - DefaultPriorMeanI, 179
 - DefaultPriorPSFSigmaAlpha, 180
 - DefaultPriorPixelMeanBG, 180
 - DefaultPriorSigmaPos, 180
 - DefaultPriorType, 180
 - DefaultProfileBoundsEstimatorMethod, 180
 - DefaultSeperableInitEstimator, 180
 - estimator_names, 181
 - eta_bg, 181
 - eta_l, 181
 - eta_x, 181
 - Gauss1DMLE, 163
 - get_hyperparam_index, 165
 - get_hyperparam_names, 165
 - get_hyperparam_value, 165
 - get_hyperparams, 165
 - get_image_from_stack, 166
 - get_lbound, 166
 - 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, 167
 - get_prior, 167
 - get_psf_sigma, 167
 - get_rng_generator, 167
 - get_rng_manager, 167
 - get_size, 168
 - get_size_image_stack, 168
 - get_stats, 168

[get_ubound](#), 168
[global_default_mcmc_sigma_scale](#), 181
[global_max_mcmc_sigma_scale](#), 181
[global_max_psf_sigma](#), 182
[global_max_size](#), 182
[global_min_psf_sigma](#), 182
[global_min_size](#), 182
[has_hyperparam](#), 168
[ImageCoordT](#), 161
[ImagePixelT](#), 161
[ImageShapeT](#), 161
[ImageSizeShapeT](#), 161
[ImageSizeVecShapeT](#), 161
[ImageSizeVecT](#), 161
[ImageSizeT](#), 161
[ImageStackShapeT](#), 162
[ImageStackT](#), 162
[ImageT](#), 162
[initial_theta_estimate](#), 168, 169
[lbound](#), 182
[make_default_prior](#), 169
[make_default_prior_beta_position](#), 169
[make_default_prior_normal_position](#), 169
[make_image](#), 169
[make_image_stack](#), 169
[make_param](#), 170
[make_param_mat](#), 170
[make_param_mat_stack](#), 170
[make_param_stack](#), 170, 171
[make_prior_beta_position](#), 171
[make_prior_component_intensity](#), 171
[make_prior_component_position_beta](#), 171
[make_prior_component_position_normal](#), 171
[make_prior_component_sigma](#), 172
[make_prior_normal_position](#), 172
[make_stencil](#), 172
[ModelDataStackT](#), 162
[ModelDataT](#), 162
[name](#), 182
[num_dim](#), 183
[num_hyperparams](#), 183
[num_params](#), 183
[num_phases](#), 183
[operator=](#), 173
[ParamVecT](#), 162
[ParamT](#), 162
[pixel_grad](#), 173
[pixel_grad2](#), 173
[pixel_hess](#), 173
[pixel_hess_update](#), 173
[pixel_model_value](#), 174
[prior](#), 183
[prior_types](#), 184
[psf_sigma](#), 184
[reflected_theta](#), 174
[reflected_theta_stack](#), 174
[rename_hyperparam](#), 174
[sample_mcmc_candidate](#), 174
[sample_prior](#), 175
[set_background_mcmc_sampling](#), 175
[set_bounds](#), 175
[set_hyperparam_names](#), 175
[set_hyperparam_value](#), 175
[set_hyperparams](#), 176
[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](#), 177
[set_prior](#), 177
[set_psf_sigma](#), 177
[set_rng_seed](#), 177
[set_size](#), 177, 178
[set_ubound](#), 178
[sigma_scale](#), 184
[size](#), 184
[StencilVecT](#), 162
[theta_in_bounds](#), 178
[theta_stack_in_bounds](#), 178
[ubound](#), 185
[mappel::Gauss1DModel](#), 185
[bound_theta](#), 192
[bounded_theta](#), 192
[bounded_theta_stack](#), 192
[bounds_epsilon](#), 206
[check_image_shape](#), 192
[check_param_shape](#), 192, 193
[check_psf_sigma](#), 193
[check_size](#), 193
[DefaultConfidenceLevel](#), 206
[DefaultEstimatorMethod](#), 207
[DefaultMCMCBurnin](#), 207
[DefaultMCMCNumSamples](#), 207
[DefaultMCMCThin](#), 207
[DefaultPriorBetaPos](#), 207
[DefaultPriorIntensityKappa](#), 207
[DefaultPriorMaxI](#), 207
[DefaultPriorMeanI](#), 207
[DefaultPriorPSFSigmaAlpha](#), 208
[DefaultPriorPixelMeanBG](#), 208
[DefaultPriorSigmaPos](#), 208
[DefaultPriorType](#), 208
[DefaultProfileBoundsEstimatorMethod](#), 208
[DefaultSeperableInitEstimator](#), 208
[eta_bg](#), 209
[eta_I](#), 209
[eta_x](#), 209

Gauss1DModel, 191
get_hyperparam_index, 193
get_hyperparam_names, 193
get_hyperparam_value, 194
get_hyperparams, 194
get_image_from_stack, 194
get_lbound, 194
get_mcmc_num_phases, 194
get_mcmc_sigma_scale, 194
get_num_hyperparams, 194
get_num_params, 195
get_num_pixels, 195
get_param_names, 195
get_prior, 195
get_psf_sigma, 195
get_rng_generator, 196
get_rng_manager, 196
get_size, 196
get_size_image_stack, 196
get_stats, 196
get_ubound, 196
global_default_mcmc_sigma_scale, 209
global_max_mcmc_sigma_scale, 209
global_max_psf_sigma, 210
global_max_size, 210
global_min_psf_sigma, 210
global_min_size, 210
has_hyperparam, 197
ImageCoordT, 190
ImagePixelT, 190
ImageShapeT, 190
ImageSizeShapeT, 190
ImageSizeVecShapeT, 190
ImageSizeVecT, 190
ImageSizeT, 190
ImageStackShapeT, 190
ImageStackT, 191
ImageT, 191
initial_theta_estimate, 197
lbound, 210
make_default_prior, 197
make_default_prior_beta_position, 197
make_default_prior_normal_position, 197
make_image, 198
make_image_stack, 198
make_param, 198
make_param_mat, 198
make_param_mat_stack, 198, 199
make_param_stack, 199
make_prior_beta_position, 199
make_prior_component_intensity, 199
make_prior_component_position_beta, 199
make_prior_component_position_normal, 200
make_prior_component_sigma, 200

make_prior_normal_position, 200
make_stencil, 200
num_dim, 210
num_hyperparams, 211
num_params, 211
num_phases, 211
operator=, 201
ParamVecT, 191
ParamT, 191
pixel_grad, 201
pixel_grad2, 201
pixel_hess, 201
pixel_hess_update, 201
pixel_model_value, 202
prior, 211
prior_types, 211
psf_sigma, 212
reflected_theta, 202
reflected_theta_stack, 202
rename_hyperparam, 202
sample_mcmc_candidate, 202
sample_prior, 203
set_background_mcmc_sampling, 203
set_bounds, 203
set_hyperparam_names, 203
set_hyperparam_value, 203
set_hyperparams, 204
set_image_in_stack, 204
set_intensity_mcmc_sampling, 204
set_lbound, 204
set_mcmc_num_phases, 204
set_mcmc_sigma_scale, 204
set_param_names, 205
set_prior, 205
set_psf_sigma, 205
set_rng_seed, 205
set_size, 205, 206
set_ubound, 206
sigma_scale, 212
size, 212
StencilVecT, 191
theta_in_bounds, 206
theta_stack_in_bounds, 206
ubound, 212
mappel::Gauss1DModel::Stencil, 859
bg, 861
compute_derivatives, 861
DXS, 862
derivatives_computed, 862
DX, 862
dx, 862
Gx, 862
I, 861
model, 863

- operator<<, [862](#)
- ParamT, [860](#)
- Stencil, [861](#)
- theta, [863](#)
- X, [863](#)
- x, [861](#)
- mappel::Gauss1DsMAP, [213](#)
 - bound_theta, [220](#)
 - bounded_theta, [220](#)
 - bounded_theta_stack, [220](#)
 - bounds_epsilon, [236](#)
 - check_image_shape, [220](#), [221](#)
 - check_param_shape, [221](#)
 - check_psf_sigma, [221](#)
 - check_size, [221](#)
 - DefaultConfidenceLevel, [236](#)
 - DefaultEstimatorMethod, [236](#)
 - DefaultMCMCBurnin, [236](#)
 - DefaultMCMCNumSamples, [236](#)
 - DefaultMCMCThin, [236](#)
 - DefaultPriorBetaPos, [236](#)
 - DefaultPriorIntensityKappa, [236](#)
 - DefaultPriorMaxI, [237](#)
 - DefaultPriorMeanI, [237](#)
 - DefaultPriorPSFSigmaAlpha, [237](#)
 - DefaultPriorPixelMeanBG, [237](#)
 - DefaultPriorSigmaPos, [237](#)
 - DefaultPriorType, [237](#)
 - DefaultProfileBoundsEstimatorMethod, [238](#)
 - DefaultSeperableInitEstimator, [238](#)
 - estimator_names, [238](#)
 - eta_bg, [238](#)
 - eta_l, [238](#)
 - eta_sigma, [238](#)
 - eta_x, [239](#)
 - Gauss1DsMAP, [219](#), [220](#)
 - get_hyperparam_index, [222](#)
 - get_hyperparam_names, [222](#)
 - get_hyperparam_value, [222](#)
 - get_hyperparams, [222](#)
 - get_image_from_stack, [222](#)
 - get_lbound, [222](#)
 - get_max_sigma, [223](#)
 - get_mcmc_num_phases, [223](#)
 - get_mcmc_sigma_scale, [223](#)
 - get_min_sigma, [223](#)
 - get_num_hyperparams, [223](#)
 - get_num_params, [223](#)
 - get_num_pixels, [223](#)
 - get_param_names, [224](#)
 - get_prior, [224](#)
 - get_rng_generator, [224](#)
 - get_rng_manager, [224](#)
 - get_size, [224](#)
 - get_size_image_stack, [225](#)
 - get_stats, [225](#)
 - get_ubound, [225](#)
 - global_default_mcmc_sigma_scale, [239](#)
 - global_max_mcmc_sigma_scale, [239](#)
 - global_max_psf_sigma, [239](#)
 - global_max_size, [239](#)
 - global_min_psf_sigma, [239](#)
 - global_min_size, [240](#)
 - has_hyperparam, [225](#)
 - ImageCoordT, [217](#)
 - ImagePixelT, [217](#)
 - ImageShapeT, [218](#)
 - ImageSizeShapeT, [218](#)
 - ImageSizeVecShapeT, [218](#)
 - ImageSizeVecT, [218](#)
 - ImageSizeT, [218](#)
 - ImageStackShapeT, [218](#)
 - ImageStackT, [218](#)
 - ImageT, [219](#)
 - initial_theta_estimate, [225](#)
 - lbound, [240](#)
 - make_default_prior, [225](#)
 - make_default_prior_beta_position, [226](#)
 - make_default_prior_normal_position, [226](#)
 - make_image, [226](#)
 - make_image_stack, [226](#)
 - make_param, [226](#)
 - make_param_mat, [227](#)
 - make_param_mat_stack, [227](#)
 - make_param_stack, [227](#)
 - make_prior_beta_position, [227](#)
 - make_prior_component_intensity, [228](#)
 - make_prior_component_position_beta, [228](#)
 - make_prior_component_position_normal, [228](#)
 - make_prior_component_sigma, [228](#)
 - make_prior_normal_position, [229](#)
 - make_stencil, [229](#)
 - ModelDataStackT, [219](#)
 - ModelDataT, [219](#)
 - name, [240](#)
 - num_dim, [240](#)
 - num_hyperparams, [240](#)
 - num_params, [240](#)
 - num_phases, [241](#)
 - operator=, [229](#), [230](#)
 - ParamVecT, [219](#)
 - ParamT, [219](#)
 - pixel_grad, [230](#)
 - pixel_grad2, [230](#)
 - pixel_hess, [230](#)
 - pixel_hess_update, [230](#)
 - pixel_model_value, [230](#)
 - prior, [241](#)

- prior_types, [241](#)
- reflected_theta, [231](#)
- reflected_theta_stack, [231](#)
- rename_hyperparam, [231](#)
- sample_mcmc_candidate, [231](#)
- sample_prior, [231](#), [232](#)
- set_background_mcmc_sampling, [232](#)
- set_bounds, [232](#)
- set_hyperparam_names, [232](#)
- set_hyperparam_value, [232](#)
- set_hyperparams, [232](#)
- set_image_in_stack, [233](#)
- set_intensity_mcmc_sampling, [233](#)
- set_lbound, [233](#)
- set_max_sigma, [233](#)
- set_mcmc_num_phases, [233](#)
- set_mcmc_sigma_scale, [234](#)
- set_min_sigma, [234](#)
- set_param_names, [234](#)
- set_prior, [234](#)
- set_rng_seed, [234](#)
- set_size, [235](#)
- set_ubound, [235](#)
- sigma_scale, [241](#)
- size, [242](#)
- StencilVecT, [219](#)
- theta_in_bounds, [235](#)
- theta_stack_in_bounds, [235](#)
- ubound, [242](#)
- mappel::Gauss1DsMLE, [242](#)
 - bound_theta, [249](#)
 - bounded_theta, [249](#)
 - bounded_theta_stack, [250](#)
 - bounds_epsilon, [265](#)
 - check_image_shape, [250](#)
 - check_param_shape, [250](#)
 - check_psf_sigma, [250](#), [251](#)
 - check_size, [251](#)
 - DefaultConfidenceLevel, [265](#)
 - DefaultEstimatorMethod, [265](#)
 - DefaultMCMCBurnin, [265](#)
 - DefaultMCMCNumSamples, [265](#)
 - DefaultMCMCThin, [266](#)
 - DefaultPriorBetaPos, [266](#)
 - DefaultPriorIntensityKappa, [266](#)
 - DefaultPriorMaxI, [266](#)
 - DefaultPriorMeanI, [266](#)
 - DefaultPriorPSFSigmaAlpha, [266](#)
 - DefaultPriorPixelMeanBG, [266](#)
 - DefaultPriorSigmaPos, [267](#)
 - DefaultPriorType, [267](#)
 - DefaultProfileBoundsEstimatorMethod, [267](#)
 - DefaultSeperableInitEstimator, [267](#)
 - estimator_names, [267](#)
 - eta_bg, [267](#)
 - eta_I, [267](#)
 - eta_sigma, [268](#)
 - eta_x, [268](#)
 - Gauss1DsMLE, [249](#)
 - get_hyperparam_index, [251](#)
 - get_hyperparam_names, [251](#)
 - get_hyperparam_value, [251](#)
 - get_hyperparams, [251](#)
 - get_image_from_stack, [252](#)
 - get_lbound, [252](#)
 - get_max_sigma, [252](#)
 - get_mcmc_num_phases, [252](#)
 - get_mcmc_sigma_scale, [252](#)
 - get_min_sigma, [252](#)
 - get_num_hyperparams, [252](#)
 - get_num_params, [253](#)
 - get_num_pixels, [253](#)
 - get_param_names, [253](#)
 - get_prior, [253](#)
 - get_rng_generator, [253](#)
 - get_rng_manager, [253](#)
 - get_size, [254](#)
 - get_size_image_stack, [254](#)
 - get_stats, [254](#)
 - get_ubound, [254](#)
 - global_default_mcmc_sigma_scale, [268](#)
 - global_max_mcmc_sigma_scale, [268](#)
 - global_max_psf_sigma, [268](#)
 - global_max_size, [268](#)
 - global_min_psf_sigma, [269](#)
 - global_min_size, [269](#)
 - has_hyperparam, [254](#)
 - ImageCoordT, [247](#)
 - ImagePixelT, [247](#)
 - ImageShapeT, [247](#)
 - ImageSizeShapeT, [247](#)
 - ImageSizeVecShapeT, [247](#)
 - ImageSizeVecT, [247](#)
 - ImageSizeT, [247](#)
 - ImageStackShapeT, [248](#)
 - ImageStackT, [248](#)
 - ImageT, [248](#)
 - initial_theta_estimate, [254](#), [255](#)
 - lbound, [269](#)
 - make_default_prior, [255](#)
 - make_default_prior_beta_position, [255](#)
 - make_default_prior_normal_position, [255](#)
 - make_image, [255](#)
 - make_image_stack, [256](#)
 - make_param, [256](#)
 - make_param_mat, [256](#)
 - make_param_mat_stack, [256](#)
 - make_param_stack, [257](#)

make_prior_beta_position, 257
make_prior_component_intensity, 257
make_prior_component_position_beta, 257
make_prior_component_position_normal, 258
make_prior_component_sigma, 258
make_prior_normal_position, 258
make_stencil, 258
ModelDataStackT, 248
ModelDataT, 248
name, 269
num_dim, 269
num_hyperparams, 269
num_params, 270
num_phases, 270
operator=, 259
ParamVecT, 248
ParamT, 248
pixel_grad, 259
pixel_grad2, 259
pixel_hess, 259
pixel_hess_update, 260
pixel_model_value, 260
prior, 270
prior_types, 270
reflected_theta, 260
reflected_theta_stack, 260
rename_hyperparam, 260
sample_mcmc_candidate, 260, 261
sample_prior, 261
set_background_mcmc_sampling, 261
set_bounds, 261
set_hyperparam_names, 261
set_hyperparam_value, 262
set_hyperparams, 262
set_image_in_stack, 262
set_intensity_mcmc_sampling, 262
set_lbound, 262
set_max_sigma, 262, 263
set_mcmc_num_phases, 263
set_mcmc_sigma_scale, 263
set_min_sigma, 263
set_param_names, 263
set_prior, 264
set_rng_seed, 264
set_size, 264
set_ubound, 264
sigma_scale, 271
size, 271
StencilVecT, 248
theta_in_bounds, 264
theta_stack_in_bounds, 265
ubound, 271
mappel::Gauss1DsModel, 272
bound_theta, 278
bounded_theta, 278
bounded_theta_stack, 279
bounds_epsilon, 294
check_image_shape, 279
check_param_shape, 279
check_psf_sigma, 279, 280
check_size, 280
DefaultConfidenceLevel, 294
DefaultEstimatorMethod, 294
DefaultMCMCBurnin, 294
DefaultMCMCNumSamples, 294
DefaultMCMCThin, 294
DefaultPriorBetaPos, 294
DefaultPriorIntensityKappa, 295
DefaultPriorMaxI, 295
DefaultPriorMeanI, 295
DefaultPriorPSFSigmaAlpha, 295
DefaultPriorPixelMeanBG, 295
DefaultPriorSigmaPos, 295
DefaultPriorType, 295
DefaultProfileBoundsEstimatorMethod, 296
DefaultSeperableInitEstimator, 296
eta_bg, 296
eta_l, 296
eta_sigma, 296
eta_x, 296
Gauss1DsModel, 278
get_hyperparam_index, 280
get_hyperparam_names, 280
get_hyperparam_value, 280
get_hyperparams, 280
get_image_from_stack, 281
get_lbound, 281
get_max_sigma, 281
get_mcmc_num_phases, 281
get_mcmc_sigma_scale, 281
get_min_sigma, 281
get_num_hyperparams, 281
get_num_params, 282
get_num_pixels, 282
get_param_names, 282
get_prior, 282
get_rng_generator, 282
get_rng_manager, 282
get_size, 283
get_size_image_stack, 283
get_stats, 283
get_ubound, 283
global_default_mcmc_sigma_scale, 297
global_max_mcmc_sigma_scale, 297
global_max_psf_sigma, 297
global_max_size, 297
global_min_psf_sigma, 297
global_min_size, 297

- [has_hyperparam](#), 283
- [ImageCoordT](#), 276
- [ImagePixelT](#), 276
- [ImageShapeT](#), 276
- [ImageSizeShapeT](#), 277
- [ImageSizeVecShapeT](#), 277
- [ImageSizeVecT](#), 277
- [ImageSizeT](#), 277
- [ImageStackShapeT](#), 277
- [ImageStackT](#), 277
- [ImageT](#), 277
- [initial_theta_estimate](#), 283, 284
- [lbound](#), 298
- [make_default_prior](#), 284
- [make_default_prior_beta_position](#), 284
- [make_default_prior_normal_position](#), 284
- [make_image](#), 284
- [make_image_stack](#), 284
- [make_param](#), 285
- [make_param_mat](#), 285
- [make_param_mat_stack](#), 285
- [make_param_stack](#), 285, 286
- [make_prior_beta_position](#), 286
- [make_prior_component_intensity](#), 286
- [make_prior_component_position_beta](#), 286
- [make_prior_component_position_normal](#), 286
- [make_prior_component_sigma](#), 287
- [make_prior_normal_position](#), 287
- [make_stencil](#), 287
- [num_dim](#), 298
- [num_hyperparams](#), 298
- [num_params](#), 298
- [num_phases](#), 298
- [operator=](#), 288
- [ParamVecT](#), 278
- [ParamT](#), 278
- [pixel_grad](#), 288
- [pixel_grad2](#), 288
- [pixel_hess](#), 288
- [pixel_hess_update](#), 288
- [pixel_model_value](#), 289
- [prior](#), 299
- [prior_types](#), 299
- [reflected_theta](#), 289
- [reflected_theta_stack](#), 289
- [rename_hyperparam](#), 289
- [sample_mcmc_candidate](#), 289
- [sample_prior](#), 290
- [set_background_mcmc_sampling](#), 290
- [set_bounds](#), 290
- [set_hyperparam_names](#), 290
- [set_hyperparam_value](#), 290
- [set_hyperparams](#), 291
- [set_image_in_stack](#), 291

- [set_intensity_mcmc_sampling](#), 291
- [set_lbound](#), 291
- [set_max_sigma](#), 291
- [set_mcmc_num_phases](#), 292
- [set_mcmc_sigma_scale](#), 292
- [set_min_sigma](#), 292
- [set_param_names](#), 292
- [set_prior](#), 292
- [set_rng_seed](#), 293
- [set_size](#), 293
- [set_ubound](#), 293
- [sigma_scale](#), 299
- [size](#), 299
- [StencilVecT](#), 278
- [theta_in_bounds](#), 293
- [theta_stack_in_bounds](#), 293
- [ubound](#), 300
- [mappel::Gauss1DsModel::Stencil](#), 845
 - [bg](#), 847
 - [compute_derivatives](#), 847
 - [DXS2](#), 848
 - [DXSX](#), 848
 - [DXS](#), 848
 - [derivatives_computed](#), 848
 - [DX](#), 848
 - [dx](#), 848
 - [Gx](#), 848
 - [I](#), 847
 - [model](#), 849
 - [operator<<](#), 848
 - [ParamT](#), 846
 - [sigma](#), 847
 - [Stencil](#), 846
 - [theta](#), 849
 - [X](#), 849
 - [x](#), 847
- [mappel::Gauss2DMAP](#), 300
 - [bound_theta](#), 308
 - [bounded_theta](#), 308
 - [bounded_theta_stack](#), 308
 - [bounds_epsilon](#), 324
 - [check_image_shape](#), 309
 - [check_param_shape](#), 309
 - [check_psf_sigma](#), 309
 - [check_size](#), 310
 - [debug_internal_sum_model_x](#), 310
 - [debug_internal_sum_model_y](#), 310
 - [DefaultConfidenceLevel](#), 324
 - [DefaultEstimatorMethod](#), 324
 - [DefaultMCMCBurnin](#), 325
 - [DefaultMCMCNumSamples](#), 325
 - [DefaultMCMCThin](#), 325
 - [DefaultPriorBetaPos](#), 325
 - [DefaultPriorIntensityKappa](#), 325

DefaultPriorMaxI, 325
DefaultPriorMeanI, 325
DefaultPriorPSFSigmaAlpha, 326
DefaultPriorPixelMeanBG, 325
DefaultPriorSigmaPos, 326
DefaultPriorType, 326
DefaultProfileBoundsEstimatorMethod, 326
DefaultSeperableInitEstimator, 326
estimator_names, 326
eta_bg, 327
eta_l, 327
eta_x, 327
eta_y, 327
Gauss1DSumModelT, 305
Gauss2DMap, 307, 308
get_hyperparam_index, 310
get_hyperparam_names, 310
get_hyperparam_value, 310
get_hyperparams, 311
get_image_from_stack, 311
get_lbound, 311
get_mcmc_num_phases, 311
get_mcmc_sigma_scale, 311
get_num_hyperparams, 311
get_num_params, 311
get_num_pixels, 312
get_param_names, 312
get_prior, 312
get_psf_sigma, 312
get_rng_generator, 312
get_rng_manager, 313
get_size, 313
get_size_image_stack, 313
get_stats, 313
get_ubound, 313
global_default_mcmc_sigma_scale, 327
global_max_mcmc_sigma_scale, 327
global_max_psf_sigma, 328
global_max_size, 328
global_min_psf_sigma, 328
global_min_size, 328
has_hyperparam, 313
ImageCoordT, 305
ImagePixelT, 305
ImageShapeT, 305
ImageSizeShapeT, 306
ImageSizeVecShapeT, 306
ImageSizeVecT, 306
ImageSizeT, 306
ImageStackShapeT, 306
ImageStackT, 306
ImageT, 306
initial_theta_estimate, 314
lbound, 328
make_default_prior, 314
make_default_prior_beta_position, 314
make_default_prior_normal_position, 314
make_image, 315
make_image_stack, 315
make_internal_1Dsum_estimator, 315
make_param, 315
make_param_mat, 316
make_param_mat_stack, 316
make_param_stack, 316
make_prior_beta_position, 316
make_prior_component_intensity, 317
make_prior_component_position_beta, 317
make_prior_component_position_normal, 317
make_prior_component_sigma, 317
make_prior_normal_position, 318
make_stencil, 318
ModelDataStackT, 307
ModelDataT, 307
name, 328
num_dim, 329
num_hyperparams, 329
num_params, 329
num_phases, 329
operator=, 318, 319
ParamVecT, 307
ParamT, 307
pixel_grad, 319
pixel_grad2, 319
pixel_hess, 319
pixel_hess_update, 319
pixel_model_value, 319
prior, 329
prior_types, 330
psf_sigma, 330
reflected_theta, 320
reflected_theta_stack, 320
rename_hyperparam, 320
sample_mcmc_candidate, 320
sample_prior, 320, 321
set_background_mcmc_sampling, 321
set_bounds, 321
set_hyperparam_names, 321
set_hyperparam_value, 321
set_hyperparams, 321
set_image_in_stack, 322
set_intensity_mcmc_sampling, 322
set_lbound, 322
set_mcmc_num_phases, 322
set_mcmc_sigma_scale, 322
set_param_names, 322
set_prior, 322, 323
set_psf_sigma, 323
set_rng_seed, 323

- set_size, 323
- set_ubound, 323
- sigma_scale, 330
- size, 330
- StencilVecT, 307
- theta_in_bounds, 323
- theta_stack_in_bounds, 324
- ubound, 331
- update_internal_1Dsum_estimators, 324
- x_model, 331
- y_model, 331
- mappel::Gauss2DMLE, 332
 - bound_theta, 339
 - bounded_theta, 339
 - bounded_theta_stack, 340
 - bounds_epsilon, 355
 - check_image_shape, 340
 - check_param_shape, 340
 - check_psf_sigma, 340, 341
 - check_size, 341
 - debug_internal_sum_model_x, 341
 - debug_internal_sum_model_y, 341
 - DefaultConfidenceLevel, 355
 - DefaultEstimatorMethod, 355
 - DefaultMCMCBurnin, 356
 - DefaultMCMCNumSamples, 356
 - DefaultMCMCThin, 356
 - DefaultPriorBetaPos, 356
 - DefaultPriorIntensityKappa, 356
 - DefaultPriorMaxI, 356
 - DefaultPriorMeanI, 356
 - DefaultPriorPSFSigmaAlpha, 357
 - DefaultPriorPixelMeanBG, 356
 - DefaultPriorSigmaPos, 357
 - DefaultPriorType, 357
 - DefaultProfileBoundsEstimatorMethod, 357
 - DefaultSeperableInitEstimator, 357
 - estimator_names, 357
 - eta_bg, 358
 - eta_l, 358
 - eta_x, 358
 - eta_y, 358
 - Gauss1DSumModelT, 337
 - Gauss2DMLE, 339
 - get_hyperparam_index, 341
 - get_hyperparam_names, 341
 - get_hyperparam_value, 342
 - get_hyperparams, 342
 - get_image_from_stack, 342
 - get_lbound, 342
 - get_mcmc_num_phases, 342
 - get_mcmc_sigma_scale, 342
 - get_num_hyperparams, 342
 - get_num_params, 343
 - get_num_pixels, 343
 - get_param_names, 343
 - get_prior, 343
 - get_psf_sigma, 343
 - get_rng_generator, 344
 - get_rng_manager, 344
 - get_size, 344
 - get_size_image_stack, 344
 - get_stats, 344
 - get_ubound, 344
 - global_default_mcmc_sigma_scale, 358
 - global_max_mcmc_sigma_scale, 358
 - global_max_psf_sigma, 359
 - global_max_size, 359
 - global_min_psf_sigma, 359
 - global_min_size, 359
 - has_hyperparam, 345
 - ImageCoordT, 337
 - ImagePixelT, 337
 - ImageShapeT, 337
 - ImageSizeShapeT, 337
 - ImageSizeVecShapeT, 337
 - ImageSizeVecT, 337
 - ImageSizeT, 337
 - ImageStackShapeT, 338
 - ImageStackT, 338
 - ImageT, 338
 - initial_theta_estimate, 345
 - lbound, 359
 - make_default_prior, 345
 - make_default_prior_beta_position, 346
 - make_default_prior_normal_position, 346
 - make_image, 346
 - make_image_stack, 346
 - make_internal_1Dsum_estimator, 346
 - make_param, 346, 347
 - make_param_mat, 347
 - make_param_mat_stack, 347
 - make_param_stack, 347
 - make_prior_beta_position, 348
 - make_prior_component_intensity, 348
 - make_prior_component_position_beta, 348
 - make_prior_component_position_normal, 348
 - make_prior_component_sigma, 348
 - make_prior_normal_position, 349
 - make_stencil, 349
 - ModelDataStackT, 338
 - ModelDataT, 338
 - name, 359
 - num_dim, 360
 - num_hyperparams, 360
 - num_params, 360
 - num_phases, 360
 - operator=, 349, 350

ParamVecT, [338](#)
 ParamT, [338](#)
 pixel_grad, [350](#)
 pixel_grad2, [350](#)
 pixel_hess, [350](#)
 pixel_hess_update, [350](#)
 pixel_model_value, [350](#)
 prior, [360](#)
 prior_types, [361](#)
 psf_sigma, [361](#)
 reflected_theta, [351](#)
 reflected_theta_stack, [351](#)
 rename_hyperparam, [351](#)
 sample_mcmc_candidate, [351](#)
 sample_prior, [351](#), [352](#)
 set_background_mcmc_sampling, [352](#)
 set_bounds, [352](#)
 set_hyperparam_names, [352](#)
 set_hyperparam_value, [352](#)
 set_hyperparams, [352](#)
 set_image_in_stack, [353](#)
 set_intensity_mcmc_sampling, [353](#)
 set_lbound, [353](#)
 set_mcmc_num_phases, [353](#)
 set_mcmc_sigma_scale, [353](#)
 set_param_names, [353](#)
 set_prior, [353](#), [354](#)
 set_psf_sigma, [354](#)
 set_rng_seed, [354](#)
 set_size, [354](#)
 set_ubound, [354](#)
 sigma_scale, [361](#)
 size, [361](#)
 StencilVecT, [338](#)
 theta_in_bounds, [354](#)
 theta_stack_in_bounds, [355](#)
 ubound, [362](#)
 update_internal_1Dsum_estimators, [355](#)
 x_model, [362](#)
 y_model, [362](#)
 mappel::Gauss2DModel, [363](#)
 bound_theta, [370](#)
 bounded_theta, [370](#)
 bounded_theta_stack, [370](#)
 bounds_epsilon, [386](#)
 check_image_shape, [371](#)
 check_param_shape, [371](#)
 check_psf_sigma, [371](#)
 check_size, [372](#)
 debug_internal_sum_model_x, [372](#)
 debug_internal_sum_model_y, [372](#)
 DefaultConfidenceLevel, [386](#)
 DefaultEstimatorMethod, [386](#)
 DefaultMCMCBurnin, [386](#)
 DefaultMCMCNumSamples, [386](#)
 DefaultMCMCThin, [386](#)
 DefaultPriorBetaPos, [386](#)
 DefaultPriorIntensityKappa, [387](#)
 DefaultPriorMaxI, [387](#)
 DefaultPriorMeanI, [387](#)
 DefaultPriorPSFSigmaAlpha, [387](#)
 DefaultPriorPixelMeanBG, [387](#)
 DefaultPriorSigmaPos, [387](#)
 DefaultPriorType, [387](#)
 DefaultProfileBoundsEstimatorMethod, [388](#)
 DefaultSeperableInitEstimator, [388](#)
 eta_bg, [388](#)
 eta_l, [388](#)
 eta_x, [388](#)
 eta_y, [388](#)
 Gauss1DSumModelT, [368](#)
 Gauss2DModel, [370](#)
 get_hyperparam_index, [372](#)
 get_hyperparam_names, [372](#)
 get_hyperparam_value, [372](#)
 get_hyperparams, [372](#)
 get_image_from_stack, [373](#)
 get_lbound, [373](#)
 get_mcmc_num_phases, [373](#)
 get_mcmc_sigma_scale, [373](#)
 get_num_hyperparams, [373](#)
 get_num_params, [373](#)
 get_num_pixels, [373](#)
 get_param_names, [374](#)
 get_prior, [374](#)
 get_psf_sigma, [374](#)
 get_rng_generator, [374](#)
 get_rng_manager, [374](#)
 get_size, [375](#)
 get_size_image_stack, [375](#)
 get_stats, [375](#)
 get_ubound, [375](#)
 global_default_mcmc_sigma_scale, [389](#)
 global_max_mcmc_sigma_scale, [389](#)
 global_max_psf_sigma, [389](#)
 global_max_size, [389](#)
 global_min_psf_sigma, [389](#)
 global_min_size, [389](#)
 has_hyperparam, [375](#)
 ImageCoordT, [368](#)
 ImagePixelT, [368](#)
 ImageShapeT, [368](#)
 ImageSizeShapeT, [368](#)
 ImageSizeVecShapeT, [369](#)
 ImageSizeVecT, [369](#)
 ImageSizeT, [368](#)
 ImageStackShapeT, [369](#)
 ImageStackT, [369](#)

ImageT, 369
initial_theta_estimate, 375, 376
lbound, 390
make_default_prior, 376
make_default_prior_beta_position, 376
make_default_prior_normal_position, 376
make_image, 376
make_image_stack, 377
make_internal_1Dsum_estimator, 377
make_param, 377
make_param_mat, 377
make_param_mat_stack, 378
make_param_stack, 378
make_prior_beta_position, 378
make_prior_component_intensity, 378
make_prior_component_position_beta, 379
make_prior_component_position_normal, 379
make_prior_component_sigma, 379
make_prior_normal_position, 379
make_stencil, 379
num_dim, 390
num_hyperparams, 390
num_params, 390
num_phases, 390
operator=, 380
ParamVecT, 369
ParamT, 369
pixel_grad, 380
pixel_grad2, 380
pixel_hess, 381
pixel_hess_update, 381
pixel_model_value, 381
prior, 391
prior_types, 391
psf_sigma, 391
reflected_theta, 381
reflected_theta_stack, 381
rename_hyperparam, 381
sample_mcmc_candidate, 382
sample_prior, 382
set_background_mcmc_sampling, 382
set_bounds, 382
set_hyperparam_names, 383
set_hyperparam_value, 383
set_hyperparams, 383
set_image_in_stack, 383
set_intensity_mcmc_sampling, 383
set_lbound, 383
set_mcmc_num_phases, 384
set_mcmc_sigma_scale, 384
set_param_names, 384
set_prior, 384
set_psf_sigma, 384
set_rng_seed, 385
set_size, 385
set_ubound, 385
sigma_scale, 391
size, 392
StencilVecT, 369
theta_in_bounds, 385
theta_stack_in_bounds, 385
ubound, 392
update_internal_1Dsum_estimators, 385
x_model, 392
y_model, 393
mappel::Gauss2DModel::Stencil, 849
bg, 851
compute_derivatives, 851
DXS, 852
DYS, 853
derivatives_computed, 852
DX, 852
dx, 852
DY, 853
dy, 852
Gx, 853
Gy, 853
I, 851
model, 853
operator<<, 852
ParamT, 850
Stencil, 850
theta, 853
X, 853
x, 851
Y, 853
y, 851
mappel::Gauss2DsMAP, 393
bound_theta, 401
bounded_theta, 401
bounded_theta_stack, 401
bounds_epsilon, 419
check_image_shape, 402
check_param_shape, 402
check_psf_sigma, 402
check_size, 403
compute_max_sigma_ratio, 403
debug_internal_sum_model_x, 403
debug_internal_sum_model_y, 403
DefaultConfidenceLevel, 419
DefaultEstimatorMethod, 419
DefaultMCMCBurnin, 419
DefaultMCMCNumSamples, 419
DefaultMCMCThin, 419
DefaultPriorBetaPos, 419
DefaultPriorIntensityKappa, 419
DefaultPriorMaxI, 420
DefaultPriorMeanI, 420

DefaultPriorPSFSigmaAlpha, 420
 DefaultPriorPixelMeanBG, 420
 DefaultPriorSigmaPos, 420
 DefaultPriorType, 420
 DefaultProfileBoundsEstimatorMethod, 421
 DefaultSeperableInitEstimator, 421
 estimator_names, 421
 eta_bg, 421
 eta_l, 421
 eta_sigma, 421
 eta_x, 422
 eta_y, 422
 Gauss1DSumModelT, 398
 Gauss2DsMAP, 400, 401
 get_hyperparam_index, 403
 get_hyperparam_names, 403
 get_hyperparam_value, 404
 get_hyperparams, 404
 get_image_from_stack, 404
 get_lbound, 404
 get_max_sigma, 404
 get_max_sigma_ratio, 405
 get_mcmc_num_phases, 405
 get_mcmc_sigma_scale, 405
 get_min_sigma, 405
 get_num_hyperparams, 405
 get_num_params, 405
 get_num_pixels, 406
 get_param_names, 406
 get_prior, 406
 get_rng_generator, 406
 get_rng_manager, 406
 get_size, 406, 407
 get_size_image_stack, 407
 get_stats, 407
 get_ubound, 407
 global_default_mcmc_sigma_scale, 422
 global_max_mcmc_sigma_scale, 422
 global_max_psf_sigma, 422
 global_max_size, 422
 global_min_psf_sigma, 423
 global_min_size, 423
 has_hyperparam, 407
 ImageCoordT, 398
 ImagePixelT, 398
 ImageShapeT, 399
 ImageSizeShapeT, 399
 ImageSizeVecShapeT, 399
 ImageSizeVecT, 399
 ImageSizeT, 399
 ImageStackShapeT, 399
 ImageStackT, 399
 ImageT, 400
 initial_theta_estimate, 407, 408
 lbound, 423
 make_default_prior, 408
 make_default_prior_beta_position, 408
 make_default_prior_normal_position, 408
 make_image, 408
 make_image_stack, 409
 make_internal_1Dsum_estimator, 409
 make_param, 409
 make_param_mat, 409
 make_param_mat_stack, 410
 make_param_stack, 410
 make_prior_beta_position, 410
 make_prior_component_intensity, 410
 make_prior_component_position_beta, 411
 make_prior_component_position_normal, 411
 make_prior_component_sigma, 411
 make_prior_normal_position, 411
 make_stencil, 412
 min_sigma, 423
 ModelDataStackT, 400
 ModelDataT, 400
 name, 423
 num_dim, 424
 num_hyperparams, 424
 num_params, 424
 num_phases, 424
 operator=, 412
 ParamVecT, 400
 ParamT, 400
 pixel_grad, 413
 pixel_grad2, 413
 pixel_hess, 413
 pixel_hess_update, 413
 pixel_model_value, 413
 prior, 424
 prior_types, 425
 reflected_theta, 414
 reflected_theta_stack, 414
 rename_hyperparam, 414
 sample_mcmc_candidate, 414
 sample_prior, 414, 415
 set_background_mcmc_sampling, 415
 set_bounds, 415
 set_hyperparam_names, 415
 set_hyperparam_value, 415
 set_hyperparams, 415
 set_image_in_stack, 416
 set_intensity_mcmc_sampling, 416
 set_lbound, 416
 set_max_sigma, 416
 set_max_sigma_ratio, 416
 set_mcmc_num_phases, 416
 set_mcmc_sigma_scale, 417
 set_min_sigma, 417

- set_param_names, 417
- set_prior, 417
- set_rng_seed, 417
- set_size, 417
- set_ubound, 418
- sigma_scale, 425
- size, 425
- StencilVecT, 400
- theta_in_bounds, 418
- theta_stack_in_bounds, 418
- ubound, 426
- update_internal_1Dsum_estimators, 418
- x_model, 426
- y_model, 426
- mappel::Gauss2DsMLE, 427
 - bound_theta, 435
 - bounded_theta, 435
 - bounded_theta_stack, 435
 - bounds_epsilon, 452
 - check_image_shape, 435
 - check_param_shape, 435, 436
 - check_psf_sigma, 436
 - check_size, 436
 - compute_max_sigma_ratio, 436
 - debug_internal_sum_model_x, 437
 - debug_internal_sum_model_y, 437
 - DefaultConfidenceLevel, 452
 - DefaultEstimatorMethod, 452
 - DefaultMCMCBurnin, 452
 - DefaultMCMCNumSamples, 452
 - DefaultMCMCThin, 453
 - DefaultPriorBetaPos, 453
 - DefaultPriorIntensityKappa, 453
 - DefaultPriorMaxI, 453
 - DefaultPriorMeanI, 453
 - DefaultPriorPSFSigmaAlpha, 453
 - DefaultPriorPixelMeanBG, 453
 - DefaultPriorSigmaPos, 454
 - DefaultPriorType, 454
 - DefaultProfileBoundsEstimatorMethod, 454
 - DefaultSeperableInitEstimator, 454
 - estimator_names, 454
 - eta_bg, 454
 - eta_l, 454
 - eta_sigma, 455
 - eta_x, 455
 - eta_y, 455
 - Gauss1DSumModelT, 432
 - Gauss2DsMLE, 434
 - get_hyperparam_index, 437
 - get_hyperparam_names, 437
 - get_hyperparam_value, 437
 - get_hyperparams, 437
 - get_image_from_stack, 438
 - get_lbound, 438
 - get_max_sigma, 438
 - get_max_sigma_ratio, 438
 - get_mcmc_num_phases, 438
 - get_mcmc_sigma_scale, 438
 - get_min_sigma, 439
 - get_num_hyperparams, 439
 - get_num_params, 439
 - get_num_pixels, 439
 - get_param_names, 439
 - get_prior, 439, 440
 - get_rng_generator, 440
 - get_rng_manager, 440
 - get_size, 440
 - get_size_image_stack, 440
 - get_stats, 440
 - get_ubound, 441
 - global_default_mcmc_sigma_scale, 455
 - global_max_mcmc_sigma_scale, 455
 - global_max_psf_sigma, 456
 - global_max_size, 456
 - global_min_psf_sigma, 456
 - global_min_size, 456
 - has_hyperparam, 441
 - ImageCoordT, 432
 - ImagePixelT, 432
 - ImageShapeT, 432
 - ImageSizeShapeT, 432
 - ImageSizeVecShapeT, 432
 - ImageSizeVecT, 433
 - ImageSizeT, 432
 - ImageStackShapeT, 433
 - ImageStackT, 433
 - ImageT, 433
 - initial_theta_estimate, 441
 - lbound, 456
 - make_default_prior, 441
 - make_default_prior_beta_position, 442
 - make_default_prior_normal_position, 442
 - make_image, 442
 - make_image_stack, 442
 - make_internal_1Dsum_estimator, 442
 - make_param, 443
 - make_param_mat, 443
 - make_param_mat_stack, 443
 - make_param_stack, 444
 - make_prior_beta_position, 444
 - make_prior_component_intensity, 444
 - make_prior_component_position_beta, 444
 - make_prior_component_position_normal, 445
 - make_prior_component_sigma, 445
 - make_prior_normal_position, 445
 - make_stencil, 445
 - min_sigma, 456

ModelDataStackT, [433](#)
 ModelDataT, [433](#)
 name, [457](#)
 num_dim, [457](#)
 num_hyperparams, [457](#)
 num_params, [457](#)
 num_phases, [457](#)
 operator=, [446](#)
 ParamVecT, [434](#)
 ParamT, [433](#)
 pixel_grad, [446](#)
 pixel_grad2, [446](#)
 pixel_hess, [446](#)
 pixel_hess_update, [447](#)
 pixel_model_value, [447](#)
 prior, [458](#)
 prior_types, [458](#)
 reflected_theta, [447](#)
 reflected_theta_stack, [447](#)
 rename_hyperparam, [447](#)
 sample_mcmc_candidate, [448](#)
 sample_prior, [448](#)
 set_background_mcmc_sampling, [448](#)
 set_bounds, [448](#)
 set_hyperparam_names, [449](#)
 set_hyperparam_value, [449](#)
 set_hyperparams, [449](#)
 set_image_in_stack, [449](#)
 set_intensity_mcmc_sampling, [449](#)
 set_lbound, [449](#)
 set_max_sigma, [450](#)
 set_max_sigma_ratio, [450](#)
 set_mcmc_num_phases, [450](#)
 set_mcmc_sigma_scale, [450](#)
 set_min_sigma, [450](#)
 set_param_names, [450](#)
 set_prior, [451](#)
 set_rng_seed, [451](#)
 set_size, [451](#)
 set_ubound, [451](#)
 sigma_scale, [458](#)
 size, [458](#)
 StencilVecT, [434](#)
 theta_in_bounds, [451](#)
 theta_stack_in_bounds, [451](#)
 ubound, [459](#)
 update_internal_1Dsum_estimators, [452](#)
 x_model, [459](#)
 y_model, [459](#)
 mappel::Gauss2DsModel, [460](#)
 bound_theta, [468](#)
 bounded_theta, [468](#)
 bounded_theta_stack, [468](#)
 bounds_epsilon, [484](#)
 check_image_shape, [468](#)
 check_param_shape, [468](#)
 check_psf_sigma, [469](#)
 check_size, [469](#)
 compute_max_sigma_ratio, [469](#)
 debug_internal_sum_model_x, [469](#)
 debug_internal_sum_model_y, [469](#)
 DefaultConfidenceLevel, [484](#)
 DefaultEstimatorMethod, [484](#)
 DefaultMCMCBurnin, [484](#)
 DefaultMCMCNumSamples, [485](#)
 DefaultMCMCThin, [485](#)
 DefaultPriorBetaPos, [485](#)
 DefaultPriorIntensityKappa, [485](#)
 DefaultPriorMaxI, [485](#)
 DefaultPriorMeanI, [485](#)
 DefaultPriorPSFSigmaAlpha, [486](#)
 DefaultPriorPixelMeanBG, [485](#)
 DefaultPriorSigmaPos, [486](#)
 DefaultPriorType, [486](#)
 DefaultProfileBoundsEstimatorMethod, [486](#)
 DefaultSeperableInitEstimator, [486](#)
 eta_bg, [486](#)
 eta_I, [487](#)
 eta_sigma, [487](#)
 eta_x, [487](#)
 eta_y, [487](#)
 Gauss1DSumModelT, [465](#)
 Gauss2DsModel, [467](#)
 get_hyperparam_index, [470](#)
 get_hyperparam_names, [470](#)
 get_hyperparam_value, [470](#)
 get_hyperparams, [470](#)
 get_image_from_stack, [470](#)
 get_lbound, [470](#)
 get_max_sigma, [470, 471](#)
 get_max_sigma_ratio, [471](#)
 get_mcmc_num_phases, [471](#)
 get_mcmc_sigma_scale, [471](#)
 get_min_sigma, [471](#)
 get_num_hyperparams, [471](#)
 get_num_params, [472](#)
 get_num_pixels, [472](#)
 get_param_names, [472](#)
 get_prior, [472](#)
 get_rng_generator, [472](#)
 get_rng_manager, [472](#)
 get_size, [473](#)
 get_size_image_stack, [473](#)
 get_stats, [473](#)
 get_ubound, [473](#)
 global_default_mcmc_sigma_scale, [487](#)
 global_max_mcmc_sigma_scale, [488](#)
 global_max_psf_sigma, [488](#)

global_max_size, 488
global_min_psf_sigma, 488
global_min_size, 488
has_hyperparam, 473
ImageCoordT, 465
ImagePixelT, 465
ImageShapeT, 465
ImageSizeShapeT, 466
ImageSizeVecShapeT, 466
ImageSizeVecT, 466
ImageSizeT, 466
ImageStackShapeT, 466
ImageStackT, 466
ImageT, 466
initial_theta_estimate, 473, 474
lbound, 488
make_default_prior, 474
make_default_prior_beta_position, 474
make_default_prior_normal_position, 474
make_image, 475
make_image_stack, 475
make_internal_1Dsum_estimator, 475
make_param, 475
make_param_mat, 475, 476
make_param_mat_stack, 476
make_param_stack, 476
make_prior_beta_position, 476
make_prior_component_intensity, 476
make_prior_component_position_beta, 477
make_prior_component_position_normal, 477
make_prior_component_sigma, 477
make_prior_normal_position, 477
make_stencil, 477
min_sigma, 489
num_dim, 489
num_hyperparams, 489
num_params, 489
num_phases, 489
operator=, 478
ParamVecT, 467
ParamT, 467
pixel_grad, 478
pixel_grad2, 478
pixel_hess, 479
pixel_hess_update, 479
pixel_model_value, 479
prior, 490
prior_types, 490
reflected_theta, 479
reflected_theta_stack, 479
rename_hyperparam, 480
sample_mcmc_candidate, 480
sample_prior, 480
set_background_mcmc_sampling, 480
set_bounds, 481
set_hyperparam_names, 481
set_hyperparam_value, 481
set_hyperparams, 481
set_image_in_stack, 481
set_intensity_mcmc_sampling, 481
set_lbound, 482
set_max_sigma, 482
set_max_sigma_ratio, 482
set_mcmc_num_phases, 482
set_mcmc_sigma_scale, 482
set_min_sigma, 482
set_param_names, 483
set_prior, 483
set_rng_seed, 483
set_size, 483
set_ubound, 483
sigma_scale, 490
size, 490
StencilVecT, 467
theta_in_bounds, 483
theta_stack_in_bounds, 484
ubound, 491
update_internal_1Dsum_estimators, 484
x_model, 491
y_model, 491
mappel::Gauss2DsModel::Stencil, 854
bg, 855
compute_derivatives, 855
DXS2, 857
DXSX, 858
DXS, 857
DYS2, 858
DYSY, 858
DYS, 858
derivatives_computed, 857
DX, 857
dx, 857
DY, 858
dy, 858
Gx, 858
Gy, 859
I, 856
model, 859
operator<<, 857
ParamT, 855
sigma_ratio, 856
sigmaX, 856
sigmaY, 856
Stencil, 855
theta, 859
X, 859
x, 856
Y, 859

- y, 856
- mappel::Gauss2DsxyMAP, 492
 - bound_theta, 498
 - bounded_theta, 498
 - bounded_theta_stack, 499
 - bounds_epsilon, 513
 - check_image_shape, 499
 - check_param_shape, 499
 - check_psf_sigma, 499, 500
 - check_size, 500
 - DefaultConfidenceLevel, 513
 - DefaultEstimatorMethod, 513
 - DefaultMCMCBurnin, 513
 - DefaultMCMCNumSamples, 513
 - DefaultMCMCThin, 514
 - DefaultPriorBetaPos, 514
 - DefaultPriorIntensityKappa, 514
 - DefaultPriorMaxI, 514
 - DefaultPriorMeanI, 514
 - DefaultPriorPSFSigmaAlpha, 514
 - DefaultPriorPixelMeanBG, 514
 - DefaultPriorSigmaPos, 515
 - DefaultPriorType, 515
 - DefaultProfileBoundsEstimatorMethod, 515
 - DefaultSeperableInitEstimator, 515
 - estimator_names, 515
 - eta_bg, 515
 - eta_I, 515
 - eta_x, 516
 - Gauss2DsxyMAP, 498
 - get_hyperparam_index, 500
 - get_hyperparam_names, 500
 - get_hyperparam_value, 500
 - get_hyperparams, 500
 - get_image_from_stack, 501
 - get_lbound, 501
 - get_mcmc_num_phases, 501
 - get_mcmc_sigma_scale, 501
 - get_num_hyperparams, 501
 - get_num_params, 501
 - get_num_pixels, 501
 - get_param_names, 502
 - get_prior, 502
 - get_psf_sigma, 502
 - get_rng_generator, 502
 - get_rng_manager, 502
 - get_size, 503
 - get_size_image_stack, 503
 - get_stats, 503
 - get_ubound, 503
 - global_default_mcmc_sigma_scale, 516
 - global_max_mcmc_sigma_scale, 516
 - global_max_psf_sigma, 516
 - global_max_size, 516
 - global_min_psf_sigma, 516
 - global_min_size, 517
 - has_hyperparam, 503
 - ImageCoordT, 496
 - ImagePixelT, 496
 - ImageShapeT, 496
 - ImageSizeShapeT, 496
 - ImageSizeVecShapeT, 497
 - ImageSizeVecT, 497
 - ImageSizeT, 497
 - ImageStackShapeT, 497
 - ImageStackT, 497
 - ImageT, 497
 - initial_theta_estimate, 503, 504
 - lbound, 517
 - make_default_prior, 504
 - make_default_prior_beta_position, 504
 - make_default_prior_normal_position, 504
 - make_image, 504
 - make_image_stack, 504
 - make_param, 505
 - make_param_mat, 505
 - make_param_mat_stack, 505
 - make_param_stack, 505, 506
 - make_prior_beta_position, 506
 - make_prior_component_intensity, 506
 - make_prior_component_position_beta, 506
 - make_prior_component_position_normal, 506
 - make_prior_component_sigma, 507
 - make_prior_normal_position, 507
 - make_stencil, 507
 - ModelDataStackT, 497
 - ModelDataT, 498
 - name, 517
 - num_dim, 517
 - num_hyperparams, 517
 - num_params, 517
 - num_phases, 518
 - ParamVecT, 498
 - ParamT, 498
 - pixel_grad, 508
 - pixel_grad2, 508
 - pixel_hess, 508
 - pixel_hess_update, 508
 - pixel_model_value, 508
 - prior, 518
 - prior_types, 518
 - psf_sigma, 518
 - reflected_theta, 508
 - reflected_theta_stack, 509
 - rename_hyperparam, 509
 - sample_mcmc_candidate, 509
 - sample_prior, 509
 - set_background_mcmc_sampling, 510

- set_bounds, 510
- set_hyperparam_names, 510
- set_hyperparam_value, 510
- set_hyperparams, 510
- set_image_in_stack, 510
- set_intensity_mcmc_sampling, 511
- set_lbound, 511
- set_mcmc_num_phases, 511
- set_mcmc_sigma_scale, 511
- set_param_names, 511
- set_prior, 511
- set_psf_sigma, 512
- set_rng_seed, 512
- set_size, 512
- set_ubound, 512
- sigma_scale, 519
- size, 519
- StencilVecT, 498
- theta_in_bounds, 512
- theta_stack_in_bounds, 513
- ubound, 519
- mappel::Gauss2DsxyModel, 520
 - bound_theta, 526
 - bounded_theta, 526
 - bounded_theta_stack, 526
 - bounds_epsilon, 538
 - check_image_shape, 526, 527
 - check_param_shape, 527
 - check_psf_sigma, 527
 - check_size, 527
 - compute_max_sigma_ratio, 528
 - DefaultConfidenceLevel, 538
 - DefaultEstimatorMethod, 539
 - DefaultMCMCBurnin, 539
 - DefaultMCMCNumSamples, 539
 - DefaultMCMCThin, 539
 - DefaultPriorBetaPos, 539
 - DefaultPriorIntensityKappa, 539
 - DefaultPriorMaxI, 539
 - DefaultPriorMeanI, 539
 - DefaultPriorPSFSigmaAlpha, 540
 - DefaultPriorPixelMeanBG, 540
 - DefaultPriorSigmaPos, 540
 - DefaultProfileBoundsEstimatorMethod, 540
 - DefaultSeperableInitEstimator, 540
 - Gauss2DsxyModel, 526
 - get_hyperparam_index, 528
 - get_hyperparam_names, 528
 - get_hyperparam_value, 528
 - get_hyperparams, 528
 - get_image_from_stack, 528
 - get_lbound, 528
 - get_max_sigma, 529
 - get_max_sigma_ratio, 529
 - get_min_sigma, 529
 - get_num_hyperparams, 529
 - get_num_params, 529
 - get_num_pixels, 529
 - get_param_names, 530
 - get_prior, 530
 - get_rng_generator, 530
 - get_rng_manager, 530
 - get_size, 530
 - get_size_image_stack, 531
 - get_stats, 531
 - get_ubound, 531
 - global_max_psf_sigma, 540
 - global_max_size, 541
 - global_min_psf_sigma, 541
 - global_min_size, 541
 - has_hyperparam, 531
 - ImageCoordT, 524
 - ImagePixelT, 524
 - ImageShapeT, 524
 - ImageSizeShapeT, 524
 - ImageSizeVecShapeT, 525
 - ImageSizeVecT, 525
 - ImageSizeT, 525
 - ImageStackShapeT, 525
 - ImageStackT, 525
 - ImageT, 525
 - initial_theta_estimate, 531
 - lbound, 541
 - make_default_prior, 532
 - make_image, 532
 - make_image_stack, 532
 - make_param, 532
 - make_param_mat, 532
 - make_param_mat_stack, 532, 533
 - make_param_stack, 533
 - make_prior_beta_position, 533
 - make_prior_component_intensity, 533
 - make_prior_component_position_beta, 533
 - make_prior_component_position_normal, 534
 - make_prior_component_sigma, 534
 - make_prior_normal_position, 534
 - make_stencil, 534
 - mcmc_candidate_eta_sigma, 541
 - mcmc_candidate_eta_y, 541
 - min_sigma, 542
 - num_dim, 542
 - num_hyperparams, 542
 - num_params, 542
 - ParamVecT, 526
 - ParamT, 525
 - pixel_grad, 535
 - pixel_grad2, 535
 - pixel_hess, 535

- pixel_hess_update, [535](#)
- pixel_model_value, [535](#)
- prior, [542](#)
- reflected_theta, [535](#)
- reflected_theta_stack, [536](#)
- rename_hyperparam, [536](#)
- sample_mcmc_candidate, [536](#)
- sample_prior, [536](#)
- set_bounds, [536](#)
- set_hyperparam_names, [536](#)
- set_hyperparam_value, [537](#)
- set_hyperparams, [537](#)
- set_image_in_stack, [537](#)
- set_lbound, [537](#)
- set_max_sigma, [537](#)
- set_max_sigma_ratio, [537](#)
- set_min_sigma, [537](#)
- set_param_names, [537](#)
- set_prior, [537](#)
- set_rng_seed, [537](#)
- set_size, [538](#)
- set_ubound, [538](#)
- size, [543](#)
- StencilVecT, [526](#)
- theta_in_bounds, [538](#)
- theta_stack_in_bounds, [538](#)
- ubound, [543](#)
- update_internal_1D_estimators, [538](#)
- x_model, [543](#)
- y_model, [544](#)
- mappel::Gauss2DsxyModel::Stencil, [840](#)
 - bg, [842](#)
 - compute_derivatives, [842](#)
 - DXS2, [843](#)
 - DXSX, [843](#)
 - DXS, [843](#)
 - DYS2, [844](#)
 - DYSX, [844](#)
 - DYSY, [844](#)
 - DYS, [844](#)
 - derivatives_computed, [843](#)
 - DX, [843](#)
 - dx, [843](#)
 - DY, [844](#)
 - dy, [844](#)
 - Gx, [844](#)
 - Gy, [844](#)
 - I, [842](#)
 - model, [844](#)
 - operator<<, [843](#)
 - ParamT, [841](#)
 - sigmaX, [842](#)
 - sigmaY, [842](#)
 - Stencil, [842](#)
 - theta, [845](#)
 - X, [845](#)
 - x, [842](#)
 - Y, [845](#)
 - y, [843](#)
- mappel::ImageFormat1DBase, [557](#)
 - check_image_shape, [560](#)
 - check_size, [560](#)
 - get_image_from_stack, [560](#)
 - get_num_pixels, [560](#)
 - get_size, [560](#), [561](#)
 - get_size_image_stack, [561](#)
 - get_stats, [561](#)
 - global_max_size, [562](#)
 - global_min_size, [562](#)
 - ImageCoordT, [558](#)
 - ImageFormat1DBase, [560](#)
 - ImagePixelT, [558](#)
 - ImageShapeT, [558](#)
 - ImageSizeShapeT, [559](#)
 - ImageSizeVecShapeT, [559](#)
 - ImageSizeVecT, [559](#)
 - ImageSizeT, [559](#)
 - ImageStackShapeT, [559](#)
 - ImageStackT, [559](#)
 - ImageT, [559](#)
 - make_image, [561](#)
 - make_image_stack, [561](#)
 - num_dim, [562](#)
 - set_image_in_stack, [561](#)
 - set_size, [561](#), [562](#)
 - size, [562](#)
- mappel::ImageFormat2DBase, [563](#)
 - check_image_shape, [566](#)
 - check_size, [566](#)
 - get_image_from_stack, [566](#)
 - get_num_pixels, [567](#)
 - get_size, [567](#)
 - get_size_image_stack, [567](#)
 - get_stats, [567](#)
 - global_max_size, [568](#)
 - global_min_size, [568](#)
 - ImageCoordT, [564](#)
 - ImageFormat2DBase, [566](#)
 - ImagePixelT, [564](#)
 - ImageShapeT, [564](#)
 - ImageSizeShapeT, [565](#)
 - ImageSizeVecShapeT, [565](#)
 - ImageSizeVecT, [565](#)
 - ImageSizeT, [565](#)
 - ImageStackShapeT, [565](#)
 - ImageStackT, [565](#)
 - ImageT, [565](#)
 - make_image, [567](#)

- make_image_stack, 567
- num_dim, 568
- operator=, 567, 568
- set_image_in_stack, 568
- set_size, 568
- size, 569
- mappel::LogicalError, 590
 - LogicalError, 590
- mappel::MAPEstimator, 590
 - bound_theta, 594
 - bounded_theta, 594
 - bounded_theta_stack, 594
 - bounds_epsilon, 603
 - check_param_shape, 594
 - check_psf_sigma, 595
 - DefaultConfidenceLevel, 603
 - DefaultEstimatorMethod, 603
 - DefaultMCMCBurnin, 603
 - DefaultMCMCNumSamples, 603
 - DefaultMCMCThin, 603
 - DefaultPriorBetaPos, 603
 - DefaultPriorIntensityKappa, 603
 - DefaultPriorMaxI, 604
 - DefaultPriorMeanI, 604
 - DefaultPriorPSFSigmaAlpha, 604
 - DefaultPriorPixelMeanBG, 604
 - DefaultPriorSigmaPos, 604
 - DefaultProfileBoundsEstimatorMethod, 604
 - DefaultSeperableInitEstimator, 605
 - get_hyperparam_index, 595
 - get_hyperparam_names, 595
 - get_hyperparam_value, 595
 - get_hyperparams, 595
 - get_lbound, 596
 - get_num_hyperparams, 596
 - get_num_params, 596
 - get_param_names, 596
 - get_prior, 596
 - get_rng_generator, 596
 - get_rng_manager, 597
 - get_stats, 597
 - get_ubound, 597
 - global_max_psf_sigma, 605
 - global_min_psf_sigma, 605
 - has_hyperparam, 597
 - lbound, 605
 - MAPEstimator, 593, 594
 - make_param, 597
 - make_param_mat, 598
 - make_param_mat_stack, 598
 - make_param_stack, 598
 - make_prior_component_intensity, 598
 - make_prior_component_position_beta, 599
 - make_prior_component_position_normal, 599
 - make_prior_component_sigma, 599
 - num_hyperparams, 605
 - num_params, 605
 - operator=, 599, 600
 - ParamVecT, 593
 - ParamT, 593
 - prior, 606
 - reflected_theta, 600
 - reflected_theta_stack, 600
 - rename_hyperparam, 600
 - sample_prior, 600
 - set_bounds, 600
 - set_hyperparam_names, 601
 - set_hyperparam_value, 601
 - set_hyperparams, 601
 - set_lbound, 601
 - set_param_names, 601
 - set_prior, 601, 602
 - set_rng_seed, 602
 - set_ubound, 602
 - theta_in_bounds, 602
 - theta_stack_in_bounds, 602
 - ubound, 606
- mappel::MCMCAdaptor1Ds, 634
 - bound_theta, 638
 - bounded_theta, 638
 - bounded_theta_stack, 638
 - bounds_epsilon, 648
 - check_param_shape, 639
 - check_psf_sigma, 639
 - DefaultConfidenceLevel, 648
 - DefaultEstimatorMethod, 649
 - DefaultMCMCBurnin, 649
 - DefaultMCMCNumSamples, 649
 - DefaultMCMCThin, 649
 - DefaultPriorBetaPos, 649
 - DefaultPriorIntensityKappa, 649
 - DefaultPriorMaxI, 649
 - DefaultPriorMeanI, 649
 - DefaultPriorPSFSigmaAlpha, 650
 - DefaultPriorPixelMeanBG, 650
 - DefaultPriorSigmaPos, 650
 - DefaultProfileBoundsEstimatorMethod, 650
 - DefaultSeperableInitEstimator, 650
 - eta_bg, 650
 - eta_l, 651
 - eta_sigma, 651
 - eta_x, 651
 - get_hyperparam_index, 639
 - get_hyperparam_names, 639
 - get_hyperparam_value, 640
 - get_hyperparams, 640
 - get_lbound, 640
 - get_mcmc_num_phases, 640

get_mcmc_sigma_scale, 640
get_num_hyperparams, 640
get_num_params, 640
get_param_names, 641
get_prior, 641
get_rng_generator, 641
get_rng_manager, 641
get_stats, 641
get_ubound, 641
global_default_mcmc_sigma_scale, 651
global_max_mcmc_sigma_scale, 651
global_max_psf_sigma, 651
global_min_psf_sigma, 652
has_hyperparam, 642
lbound, 652
MCMCAdaptor1Ds, 638
make_param, 642
make_param_mat, 642
make_param_mat_stack, 642, 643
make_param_stack, 643
make_prior_component_intensity, 643
make_prior_component_position_beta, 643
make_prior_component_position_normal, 644
make_prior_component_sigma, 644
num_hyperparams, 652
num_params, 652
num_phases, 652
operator=, 644
ParamVecT, 637
ParamT, 637
prior, 653
reflected_theta, 644
reflected_theta_stack, 645
rename_hyperparam, 645
sample_mcmc_candidate, 645
sample_prior, 645
set_background_mcmc_sampling, 646
set_bounds, 646
set_hyperparam_names, 646
set_hyperparam_value, 646
set_hyperparams, 646
set_intensity_mcmc_sampling, 646
set_lbound, 647
set_mcmc_num_phases, 647
set_mcmc_sigma_scale, 647
set_param_names, 647
set_prior, 647
set_rng_seed, 648
set_ubound, 648
sigma_scale, 653
theta_in_bounds, 648
theta_stack_in_bounds, 648
ubound, 653
mappel::MCMCAdaptor1D, 615
bound_theta, 619
bounded_theta, 619
bounded_theta_stack, 619
bounds_epsilon, 629
check_param_shape, 620
check_psf_sigma, 620
DefaultConfidenceLevel, 629
DefaultEstimatorMethod, 629
DefaultMCMCBurnin, 630
DefaultMCMCNumSamples, 630
DefaultMCMCThin, 630
DefaultPriorBetaPos, 630
DefaultPriorIntensityKappa, 630
DefaultPriorMaxI, 630
DefaultPriorMeanI, 630
DefaultPriorPSFSigmaAlpha, 631
DefaultPriorPixelMeanBG, 630
DefaultPriorSigmaPos, 631
DefaultProfileBoundsEstimatorMethod, 631
DefaultSeperableInitEstimator, 631
eta_bg, 631
eta_I, 631
eta_x, 632
get_hyperparam_index, 620
get_hyperparam_names, 620
get_hyperparam_value, 621
get_hyperparams, 621
get_lbound, 621
get_mcmc_num_phases, 621
get_mcmc_sigma_scale, 621
get_num_hyperparams, 621
get_num_params, 621
get_param_names, 622
get_prior, 622
get_rng_generator, 622
get_rng_manager, 622
get_stats, 622
get_ubound, 622
global_default_mcmc_sigma_scale, 632
global_max_mcmc_sigma_scale, 632
global_max_psf_sigma, 632
global_min_psf_sigma, 632
has_hyperparam, 623
lbound, 632
MCMCAdaptor1D, 619
make_param, 623
make_param_mat, 623
make_param_mat_stack, 623, 624
make_param_stack, 624
make_prior_component_intensity, 624
make_prior_component_position_beta, 624
make_prior_component_position_normal, 625
make_prior_component_sigma, 625
num_hyperparams, 633

- num_params, 633
- num_phases, 633
- operator=, 625
- ParamVecT, 618
- ParamT, 618
- prior, 633
- reflected_theta, 625
- reflected_theta_stack, 626
- rename_hyperparam, 626
- sample_mcmc_candidate, 626
- sample_prior, 626
- set_background_mcmc_sampling, 626
- set_bounds, 627
- set_hyperparam_names, 627
- set_hyperparam_value, 627
- set_hyperparams, 627
- set_intensity_mcmc_sampling, 627
- set_lbound, 627
- set_mcmc_num_phases, 628
- set_mcmc_sigma_scale, 628
- set_param_names, 628
- set_prior, 628
- set_rng_seed, 628
- set_ubound, 629
- sigma_scale, 633
- theta_in_bounds, 629
- theta_stack_in_bounds, 629
- ubound, 634
- mappel::MCMCAdaptor2Ds, 673
 - bound_theta, 677
 - bounded_theta, 677
 - bounded_theta_stack, 678
 - bounds_epsilon, 687
 - check_param_shape, 678
 - check_psf_sigma, 678
 - DefaultConfidenceLevel, 687
 - DefaultEstimatorMethod, 688
 - DefaultMCMCBurnin, 688
 - DefaultMCMCNumSamples, 688
 - DefaultMCMCThin, 688
 - DefaultPriorBetaPos, 688
 - DefaultPriorIntensityKappa, 688
 - DefaultPriorMaxI, 688
 - DefaultPriorMeanI, 688
 - DefaultPriorPSFSigmaAlpha, 689
 - DefaultPriorPixelMeanBG, 689
 - DefaultPriorSigmaPos, 689
 - DefaultProfileBoundsEstimatorMethod, 689
 - DefaultSeperableInitEstimator, 689
 - eta_bg, 689
 - eta_l, 690
 - eta_sigma, 690
 - eta_x, 690
 - eta_y, 690
 - get_hyperparam_index, 678
 - get_hyperparam_names, 679
 - get_hyperparam_value, 679
 - get_hyperparams, 679
 - get_lbound, 679
 - get_mcmc_num_phases, 679
 - get_mcmc_sigma_scale, 679
 - get_num_hyperparams, 679
 - get_num_params, 680
 - get_param_names, 680
 - get_prior, 680
 - get_rng_generator, 680
 - get_rng_manager, 680
 - get_stats, 680
 - get_ubound, 680
 - global_default_mcmc_sigma_scale, 690
 - global_max_mcmc_sigma_scale, 691
 - global_max_psf_sigma, 691
 - global_min_psf_sigma, 691
 - has_hyperparam, 681
 - lbound, 691
 - MCMCAdaptor2Ds, 677
 - make_param, 681
 - make_param_mat, 681
 - make_param_mat_stack, 681, 682
 - make_param_stack, 682
 - make_prior_component_intensity, 682
 - make_prior_component_position_beta, 682
 - make_prior_component_position_normal, 683
 - make_prior_component_sigma, 683
 - num_hyperparams, 691
 - num_params, 691
 - num_phases, 692
 - operator=, 683
 - ParamVecT, 676
 - ParamT, 676
 - prior, 692
 - reflected_theta, 683
 - reflected_theta_stack, 684
 - rename_hyperparam, 684
 - sample_mcmc_candidate, 684
 - sample_prior, 684
 - set_background_mcmc_sampling, 685
 - set_bounds, 685
 - set_hyperparam_names, 685
 - set_hyperparam_value, 685
 - set_hyperparams, 685
 - set_intensity_mcmc_sampling, 685
 - set_lbound, 686
 - set_mcmc_num_phases, 686
 - set_mcmc_sigma_scale, 686
 - set_param_names, 686
 - set_prior, 686
 - set_rng_seed, 687

- set_ubound, 687
- sigma_scale, 692
- theta_in_bounds, 687
- theta_stack_in_bounds, 687
- ubound, 692
- mappel::MCMCAdaptor2D, 654
 - bound_theta, 658
 - bounded_theta, 658
 - bounded_theta_stack, 658
 - bounds_epsilon, 668
 - check_param_shape, 658
 - check_psf_sigma, 658, 659
 - DefaultConfidenceLevel, 668
 - DefaultEstimatorMethod, 668
 - DefaultMCMCBurnin, 668
 - DefaultMCMCNumSamples, 668
 - DefaultMCMCThin, 668
 - DefaultPriorBetaPos, 668
 - DefaultPriorIntensityKappa, 669
 - DefaultPriorMaxI, 669
 - DefaultPriorMeanI, 669
 - DefaultPriorPSFSigmaAlpha, 669
 - DefaultPriorPixelMeanBG, 669
 - DefaultPriorSigmaPos, 669
 - DefaultProfileBoundsEstimatorMethod, 670
 - DefaultSeperableInitEstimator, 670
 - eta_bg, 670
 - eta_l, 670
 - eta_x, 670
 - eta_y, 670
 - get_hyperparam_index, 659
 - get_hyperparam_names, 659
 - get_hyperparam_value, 659
 - get_hyperparams, 659
 - get_lbound, 659
 - get_mcmc_num_phases, 660
 - get_mcmc_sigma_scale, 660
 - get_num_hyperparams, 660
 - get_num_params, 660
 - get_param_names, 660
 - get_prior, 660
 - get_rng_generator, 661
 - get_rng_manager, 661
 - get_stats, 661
 - get_ubound, 661
 - global_default_mcmc_sigma_scale, 671
 - global_max_mcmc_sigma_scale, 671
 - global_max_psf_sigma, 671
 - global_min_psf_sigma, 671
 - has_hyperparam, 661
 - lbound, 671
 - MCMCAdaptor2D, 657
 - make_param, 661
 - make_param_mat, 662
 - make_param_mat_stack, 662
 - make_param_stack, 662
 - make_prior_component_intensity, 662
 - make_prior_component_position_beta, 663
 - make_prior_component_position_normal, 663
 - make_prior_component_sigma, 663
 - num_hyperparams, 671
 - num_params, 672
 - num_phases, 672
 - operator=, 663, 664
 - ParamVecT, 657
 - ParamT, 657
 - prior, 672
 - reflected_theta, 664
 - reflected_theta_stack, 664
 - rename_hyperparam, 664
 - sample_mcmc_candidate, 664
 - sample_prior, 665
 - set_background_mcmc_sampling, 665
 - set_bounds, 665
 - set_hyperparam_names, 665
 - set_hyperparam_value, 665
 - set_hyperparams, 666
 - set_intensity_mcmc_sampling, 666
 - set_lbound, 666
 - set_mcmc_num_phases, 666
 - set_mcmc_sigma_scale, 666
 - set_param_names, 666
 - set_prior, 667
 - set_rng_seed, 667
 - set_ubound, 667
 - sigma_scale, 672
 - theta_in_bounds, 667
 - theta_stack_in_bounds, 667
 - ubound, 673
- mappel::MCMCAdaptorBase, 693
 - get_mcmc_num_phases, 694
 - get_mcmc_sigma_scale, 694
 - get_stats, 694
 - global_default_mcmc_sigma_scale, 695
 - global_max_mcmc_sigma_scale, 695
 - MCMCAdaptorBase, 694
 - num_phases, 695
 - set_mcmc_num_phases, 694
 - set_mcmc_sigma_scale, 695
 - sigma_scale, 695
- mappel::MLEstimator, 704
 - bound_theta, 708
 - bounded_theta, 708
 - bounded_theta_stack, 708
 - bounds_epsilon, 717
 - check_param_shape, 708
 - check_psf_sigma, 708, 709
 - DefaultConfidenceLevel, 717

DefaultEstimatorMethod, 717
DefaultMCMCBurnin, 717
DefaultMCMCNumSamples, 717
DefaultMCMCThin, 717
DefaultPriorBetaPos, 717
DefaultPriorIntensityKappa, 717
DefaultPriorMaxI, 718
DefaultPriorMeanI, 718
DefaultPriorPSFSigmaAlpha, 718
DefaultPriorPixelMeanBG, 718
DefaultPriorSigmaPos, 718
DefaultProfileBoundsEstimatorMethod, 718
DefaultSeperableInitEstimator, 719
get_hyperparam_index, 709
get_hyperparam_names, 709
get_hyperparam_value, 709
get_hyperparams, 709
get_lbound, 709
get_num_hyperparams, 710
get_num_params, 710
get_param_names, 710
get_prior, 710
get_rng_generator, 710
get_rng_manager, 710
get_stats, 711
get_ubound, 711
global_max_psf_sigma, 719
global_min_psf_sigma, 719
has_hyperparam, 711
lbound, 719
MLEstimator, 707
make_param, 711
make_param_mat, 711, 712
make_param_mat_stack, 712
make_param_stack, 712
make_prior_component_intensity, 712
make_prior_component_position_beta, 713
make_prior_component_position_normal, 713
make_prior_component_sigma, 713
num_hyperparams, 719
num_params, 719
operator=, 713, 714
ParamVecT, 707
ParamT, 707
prior, 720
reflected_theta, 714
reflected_theta_stack, 714
rename_hyperparam, 714
sample_prior, 714
set_bounds, 714
set_hyperparam_names, 715
set_hyperparam_value, 715
set_hyperparams, 715
set_lbound, 715
set_param_names, 715
set_prior, 715, 716
set_rng_seed, 716
set_ubound, 716
theta_in_bounds, 716
theta_stack_in_bounds, 716
ubound, 720
mappel::ModelBoundsError, 720
 ModelBoundsError, 721
mappel::NotImplementedError, 763
 NotImplementedError, 764
mappel::NumericalError, 764
 NumericalError, 764
mappel::ParameterValueError, 767
 ParameterValueError, 767
mappel::PointEmitterModel, 768
 bound_theta, 772
 bounded_theta, 772
 bounded_theta_stack, 772
 bounds_epsilon, 780
 check_param_shape, 772
 check_psf_sigma, 772, 773
 DefaultConfidenceLevel, 780
 DefaultEstimatorMethod, 780
 DefaultMCMCBurnin, 780
 DefaultMCMCNumSamples, 781
 DefaultMCMCThin, 781
 DefaultPriorBetaPos, 781
 DefaultPriorIntensityKappa, 781
 DefaultPriorMaxI, 781
 DefaultPriorMeanI, 781
 DefaultPriorPSFSigmaAlpha, 782
 DefaultPriorPixelMeanBG, 781
 DefaultPriorSigmaPos, 782
 DefaultProfileBoundsEstimatorMethod, 782
 DefaultSeperableInitEstimator, 782
 get_hyperparam_index, 773
 get_hyperparam_names, 773
 get_hyperparam_value, 773
 get_hyperparams, 773
 get_lbound, 773
 get_num_hyperparams, 773
 get_num_params, 774
 get_param_names, 774
 get_prior, 774
 get_rng_generator, 774
 get_rng_manager, 774
 get_stats, 774
 get_ubound, 775
 global_max_psf_sigma, 782
 global_min_psf_sigma, 782
 has_hyperparam, 775
 lbound, 783
 make_param, 775

- make_param_mat, 775
- make_param_mat_stack, 776
- make_param_stack, 776
- make_prior_component_intensity, 776
- make_prior_component_position_beta, 776
- make_prior_component_position_normal, 777
- make_prior_component_sigma, 777
- num_hyperparams, 783
- num_params, 783
- operator=, 777
- ParamVecT, 771
- ParamT, 771
- PointEmitterModel, 771
- prior, 783
- reflected_theta, 777
- reflected_theta_stack, 778
- rename_hyperparam, 778
- sample_prior, 778
- set_bounds, 778
- set_hyperparam_names, 778
- set_hyperparam_value, 778
- set_hyperparams, 779
- set_lbound, 779
- set_param_names, 779
- set_prior, 779
- set_rng_seed, 779
- set_ubound, 779
- theta_in_bounds, 780
- theta_stack_in_bounds, 780
- ubound, 783
- mappel::PoissonNoise1DObjective, 786
 - check_image_shape, 790
 - check_size, 790
 - estimator_names, 792
 - get_image_from_stack, 790
 - get_num_pixels, 790
 - get_size, 790, 791
 - get_size_image_stack, 791
 - get_stats, 791
 - global_max_size, 792
 - global_min_size, 792
 - ImageCoordT, 788
 - ImagePixelT, 788
 - ImageShapeT, 788
 - ImageSizeShapeT, 788
 - ImageSizeVecShapeT, 788
 - ImageSizeVecT, 788
 - ImageSizeT, 788
 - ImageStackShapeT, 789
 - ImageStackT, 789
 - ImageT, 789
 - make_image, 791
 - make_image_stack, 791
 - ModelDataStackT, 789
- ModelDataT, 789
- num_dim, 792
- operator=, 791
- PoissonNoise1DObjective, 789, 790
- set_image_in_stack, 792
- set_size, 792
- size, 793
- mappel::PoissonNoise2DObjective, 793
 - check_image_shape, 797
 - check_size, 797
 - estimator_names, 799
 - get_image_from_stack, 797
 - get_num_pixels, 798
 - get_size, 798
 - get_size_image_stack, 798
 - get_stats, 798
 - global_max_size, 799
 - global_min_size, 799
 - ImageCoordT, 795
 - ImagePixelT, 795
 - ImageShapeT, 795
 - ImageSizeShapeT, 795
 - ImageSizeVecShapeT, 796
 - ImageSizeVecT, 796
 - ImageSizeT, 795
 - ImageStackShapeT, 796
 - ImageStackT, 796
 - ImageT, 796
 - make_image, 798
 - make_image_stack, 798
 - ModelDataStackT, 796
 - ModelDataT, 796
 - num_dim, 800
 - operator=, 799
 - PoissonNoise2DObjective, 797
 - set_image_in_stack, 799
 - set_size, 799
 - size, 800
- mappel::estimator, 47
 - Error, 50
 - ExitCode, 49
 - FunctionValue, 50
 - GradRatio, 50
 - MaxBacktracks, 50
 - MaxIter, 50
 - ModelImprovement, 50
 - operator<, 50
 - StepSize, 50
 - Success, 50
 - TrustRegionRadius, 50
 - Unassigned, 50
- mappel::estimator::CGaussHeuristicEstimator
 - CGaussHeuristicEstimator, 93
 - clear_stats, 93

- compute_estimate_debug, 93
- compute_profile_bound, 94
- compute_profile_bound_debug, 94
- compute_profile_estimate, 94
- estimate_max, 94, 95
- estimate_max_debug, 96
- estimate_max_stack, 97
- estimate_profile_bounds, 98
- estimate_profile_bounds_debug, 98
- estimate_profile_bounds_parallel, 99
- estimate_profile_bounds_stack, 99
- estimate_profile_max, 100
- exit_counts, 102
- get_debug_stats, 100
- get_exit_counts, 100
- get_model, 101
- get_stats, 101
- max_threads, 102
- model, 102
- mtx, 102
- name, 101
- num_estimations, 103
- num_threads, 103
- record_exit_code, 101
- record_walltime, 101
- total_walltime, 103
- mappel::estimator::CGaussHeuristicEstimator< Model >, 92
- mappel::estimator::CGaussMLE< Model >, 104
- mappel::estimator::CGaussMLE
 - CGaussMLE, 105
 - clear_stats, 106
 - compute_profile_bound, 106
 - compute_profile_bound_debug, 106
 - compute_profile_estimate, 106
 - DefaultIterations, 114
 - estimate_max, 106, 107
 - estimate_max_debug, 108
 - estimate_max_stack, 109
 - estimate_profile_bounds, 110
 - estimate_profile_bounds_debug, 110
 - estimate_profile_bounds_parallel, 111
 - estimate_profile_bounds_stack, 111
 - estimate_profile_max, 112
 - exit_counts, 114
 - get_debug_stats, 112
 - get_exit_counts, 113
 - get_model, 113
 - get_stats, 113
 - max_threads, 114
 - model, 115
 - mtx, 115
 - name, 113
 - num_estimations, 115
 - num_threads, 115
 - record_exit_code, 113
 - record_walltime, 114
 - total_walltime, 116
- mappel::estimator::Estimator
 - ~Estimator, 118
 - clear_stats, 118
 - compute_estimate, 118
 - compute_estimate_debug, 119
 - compute_profile_bound, 119
 - compute_profile_bound_debug, 119
 - compute_profile_estimate, 119
 - estimate_max, 120, 121
 - estimate_max_debug, 121, 122
 - estimate_max_stack, 122, 123
 - estimate_profile_bounds, 123
 - estimate_profile_bounds_debug, 123
 - estimate_profile_bounds_parallel, 124
 - estimate_profile_bounds_stack, 124
 - estimate_profile_max, 124
 - Estimator, 118
 - exit_counts, 126
 - get_debug_stats, 124
 - get_exit_counts, 125
 - get_model, 125
 - get_stats, 125
 - model, 126
 - name, 125
 - num_estimations, 127
 - operator<<, 126
 - record_exit_code, 125
 - record_walltime, 126
 - total_walltime, 127
- mappel::estimator::Estimator< Model >, 116
- mappel::estimator::HeuristicEstimator
 - clear_stats, 546
 - compute_estimate_debug, 546
 - compute_profile_bound, 546
 - compute_profile_bound_debug, 547
 - compute_profile_estimate, 547
 - estimate_max, 547, 548
 - estimate_max_debug, 549
 - estimate_max_stack, 550
 - estimate_profile_bounds, 551
 - estimate_profile_bounds_debug, 551
 - estimate_profile_bounds_parallel, 551
 - estimate_profile_bounds_stack, 552
 - estimate_profile_max, 552, 553
 - exit_counts, 555
 - get_debug_stats, 553
 - get_exit_counts, 553
 - get_model, 553
 - get_stats, 554
 - HeuristicEstimator, 546

- max_threads, 555
- model, 555
- mtx, 555
- name, 554
- num_estimations, 556
- num_threads, 556
- record_exit_code, 554
- record_walltime, 554
- total_walltime, 556
- mappel::estimator::HeuristicEstimator< Model >, 544
- mappel::estimator::IterativeMaximizer
 - backtrack, 572
 - backtrack_max_ratio, 585
 - backtrack_min_linear_step_ratio, 585
 - backtrack_min_ratio, 586
 - clear_stats, 572
 - compute_estimate, 573
 - compute_estimate_debug, 573
 - compute_profile_bound, 573
 - compute_profile_bound_debug, 574
 - compute_profile_estimate, 574
 - convergence_min_function_change_ratio, 586
 - convergence_min_step_size_ratio, 586
 - convergence_test_grad_ratio, 574
 - convergence_test_step_size, 575
 - DefaultIterations, 586
 - estimate_max, 575, 576
 - estimate_max_debug, 576, 577
 - estimate_max_stack, 577, 578
 - estimate_profile_bounds, 578
 - estimate_profile_bounds_debug, 579
 - estimate_profile_bounds_parallel, 579
 - estimate_profile_bounds_stack, 579
 - estimate_profile_max, 580
 - exit_counts, 586
 - get_debug_stats, 580
 - get_exit_counts, 581
 - get_model, 581
 - get_stats, 581
 - get_total_backtracks, 581
 - get_total_der_evals, 582
 - get_total_fun_evals, 582
 - get_total_iterations, 582
 - IterativeMaximizer, 572
 - last_backtrack_idx, 586
 - local_maximize, 582
 - local_profile_maximize, 582
 - max_backtracks, 587
 - max_iterations, 587
 - max_threads, 587
 - maximize, 583
 - mean_backtracks, 583
 - mean_der_evals, 584
 - mean_fun_evals, 584
 - mean_iterations, 584
 - min_eigenvalue_correction_delta, 587
 - min_profile_bound_residual, 587
 - model, 587
 - mtx, 588
 - name, 584
 - num_estimations, 588
 - num_threads, 588
 - profile_bound_backtrack, 584
 - record_exit_code, 584
 - record_run_statistics, 584
 - record_walltime, 585
 - solve_profile_bound, 585
 - total_backtracks, 588
 - total_der_evals, 589
 - total_fun_evals, 589
 - total_iterations, 589
 - total_walltime, 589
- mappel::estimator::IterativeMaximizer< Model >, 569
- mappel::estimator::IterativeMaximizer< Model >::
 - MaximizerData, 606
- mappel::estimator::IterativeMaximizer::MaximizerData
 - backtrack_idx, 612
 - current_stencil, 612
 - DefaultMaxSeqLength, 613
 - expand_max_seq_len, 609
 - fixed_idx, 613
 - free_idx, 613
 - get_backtrack_idx, 609
 - get_sequence_len, 609
 - get_theta_sequence, 609
 - get_theta_sequence_rllh, 609
 - grad, 613
 - has_fixed_parameters, 609
 - has_theta_sequence, 609
 - im, 613
 - max_seq_len, 613
 - MaximizerData, 608
 - nBacktracks, 613
 - nIterations, 614
 - num_fixed_parameters, 610
 - num_params, 614
 - record_backtrack, 610
 - record_iteration, 610
 - restore_stencil, 611
 - rllh, 614
 - s0, 614
 - s1, 614
 - save_stencil, 611
 - saved_stencil, 611
 - saved_theta, 611
 - seq_len, 614
 - seq_rllh, 615
 - set_fixed_parameters, 611

- set_stencil, 612
- stencil, 612
- step, 615
- theta, 612
- theta_seq, 615
- mappel::estimator::MLEData, 48, 903
- mappel::estimator::MLEDataStack, 48, 903
- mappel::estimator::MLEDebugData, 702
 - makeMLEData, 702
 - Nseq, 702
 - obsI, 702
 - rllh, 703
 - sequence, 703
 - sequence_rllh, 703
 - theta, 703
- mappel::estimator::NewtonDiagonalMaximizer
 - backtrack, 725
 - backtrack_max_ratio, 738
 - backtrack_min_linear_step_ratio, 738
 - backtrack_min_ratio, 738
 - clear_stats, 725
 - compute_estimate, 725
 - compute_estimate_debug, 725
 - compute_profile_bound, 726
 - compute_profile_bound_debug, 726
 - compute_profile_estimate, 726
 - convergence_min_function_change_ratio, 738
 - convergence_min_step_size_ratio, 738
 - convergence_test_grad_ratio, 727
 - convergence_test_step_size, 727
 - DefaultIterations, 739
 - estimate_max, 727, 728
 - estimate_max_debug, 729
 - estimate_max_stack, 730
 - estimate_profile_bounds, 731
 - estimate_profile_bounds_debug, 731
 - estimate_profile_bounds_parallel, 731
 - estimate_profile_bounds_stack, 731
 - estimate_profile_max, 732
 - exit_counts, 739
 - get_debug_stats, 732
 - get_exit_counts, 733
 - get_model, 733
 - get_stats, 733
 - get_total_backtracks, 733
 - get_total_der_evals, 734
 - get_total_fun_evals, 734
 - get_total_iterations, 734
 - last_backtrack_idx, 739
 - local_maximize, 734
 - local_profile_maximize, 734
 - max_backtracks, 739
 - max_iterations, 739
 - max_threads, 739
 - maximize, 735
 - MaximizerData, 724
 - mean_backtracks, 735
 - mean_der_evals, 736
 - mean_fun_evals, 736
 - mean_iterations, 736
 - min_eigenvalue_correction_delta, 740
 - min_profile_bound_residual, 740
 - model, 740
 - mtx, 740
 - name, 736
 - NewtonDiagonalMaximizer, 724
 - num_estimations, 740
 - num_threads, 741
 - profile_bound_backtrack, 736
 - record_exit_code, 736
 - record_run_statistics, 737
 - record_walltime, 737
 - solve_profile_bound, 737
 - total_backtracks, 741
 - total_der_evals, 741
 - total_fun_evals, 741
 - total_iterations, 741
 - total_walltime, 742
- mappel::estimator::NewtonDiagonalMaximizer< Model >, 721
- mappel::estimator::NewtonMaximizer
 - backtrack, 746
 - backtrack_max_ratio, 759
 - backtrack_min_linear_step_ratio, 759
 - backtrack_min_ratio, 759
 - clear_stats, 746
 - compute_estimate, 746
 - compute_estimate_debug, 746
 - compute_profile_bound, 747
 - compute_profile_bound_debug, 747
 - compute_profile_estimate, 747
 - convergence_min_function_change_ratio, 759
 - convergence_min_step_size_ratio, 759
 - convergence_test_grad_ratio, 748
 - convergence_test_step_size, 748
 - DefaultIterations, 760
 - estimate_max, 748, 749
 - estimate_max_debug, 750
 - estimate_max_stack, 751
 - estimate_profile_bounds, 752
 - estimate_profile_bounds_debug, 752
 - estimate_profile_bounds_parallel, 752
 - estimate_profile_bounds_stack, 752
 - estimate_profile_max, 753
 - exit_counts, 760
 - get_debug_stats, 753
 - get_exit_counts, 754
 - get_model, 754

- get_stats, 754
- get_total_backtracks, 754
- get_total_der_evals, 755
- get_total_fun_evals, 755
- get_total_iterations, 755
- last_backtrack_idx, 760
- local_maximize, 755
- local_profile_maximize, 755
- max_backtracks, 760
- max_iterations, 760
- max_threads, 760
- maximize, 756
- MaximizerData, 745
- mean_backtracks, 756
- mean_der_evals, 757
- mean_fun_evals, 757
- mean_iterations, 757
- min_eigenvalue_correction_delta, 761
- min_profile_bound_residual, 761
- model, 761
- mtx, 761
- name, 757
- NewtonMaximizer, 745
- num_estimations, 761
- num_threads, 762
- profile_bound_backtrack, 757
- record_exit_code, 757
- record_run_statistics, 758
- record_walltime, 758
- solve_profile_bound, 758
- total_backtracks, 762
- total_der_evals, 762
- total_fun_evals, 762
- total_iterations, 762
- total_walltime, 763
- mappel::estimator::NewtonMaximizer< Model >, 742
- mappel::estimator::ProfileBoundsData, 801
 - confidence, 802
 - estimated_idx, 802
 - initialize_arrays, 801
 - mle, 802
 - Nparams_est, 802
 - profile_lb, 802
 - profile_points_lb, 802
 - profile_points_lb_rllh, 803
 - profile_points_ub, 803
 - profile_points_ub_rllh, 803
 - profile_ub, 803
 - target_rllh_delta, 803
- mappel::estimator::ProfileBoundsDataStack, 804
 - confidence, 805
 - estimated_idx, 805
 - initialize_arrays, 805
 - mle, 805
- Ndata, 805
- Nparams_est, 805
- profile_lb, 806
- profile_points_lb, 806
- profile_points_lb_rllh, 806
- profile_points_ub, 806
- profile_points_ub_rllh, 806
- profile_ub, 806
- target_rllh_delta, 807
- mappel::estimator::ProfileBoundsDebugData, 48, 904
- mappel::estimator::ProfileLikelihoodData, 49, 904
- mappel::estimator::QuasiNewtonMaximizer
 - backtrack, 811
 - backtrack_max_ratio, 824
 - backtrack_min_linear_step_ratio, 824
 - backtrack_min_ratio, 824
 - clear_stats, 811
 - compute_estimate, 811
 - compute_estimate_debug, 811
 - compute_profile_bound, 812
 - compute_profile_bound_debug, 812
 - compute_profile_estimate, 812
 - convergence_min_function_change_ratio, 824
 - convergence_min_step_size_ratio, 824
 - convergence_test_grad_ratio, 813
 - convergence_test_step_size, 813
 - DefaultIterations, 825
 - estimate_max, 813, 814
 - estimate_max_debug, 815
 - estimate_max_stack, 816
 - estimate_profile_bounds, 817
 - estimate_profile_bounds_debug, 817
 - estimate_profile_bounds_parallel, 817
 - estimate_profile_bounds_stack, 817
 - estimate_profile_max, 818
 - exit_counts, 825
 - get_debug_stats, 818
 - get_exit_counts, 819
 - get_model, 819
 - get_stats, 819
 - get_total_backtracks, 819
 - get_total_der_evals, 820
 - get_total_fun_evals, 820
 - get_total_iterations, 820
 - last_backtrack_idx, 825
 - local_maximize, 820
 - local_profile_maximize, 820
 - max_backtracks, 825
 - max_iterations, 825
 - max_threads, 825
 - maximize, 821
 - MaximizerData, 810
 - mean_backtracks, 821
 - mean_der_evals, 822

- mean_fun_evals, [822](#)
- mean_iterations, [822](#)
- min_eigenvalue_correction_delta, [826](#)
- min_profile_bound_residual, [826](#)
- model, [826](#)
- mtx, [826](#)
- name, [822](#)
- num_estimations, [826](#)
- num_threads, [827](#)
- profile_bound_backtrack, [822](#)
- QuasiNewtonMaximizer, [810](#)
- record_exit_code, [822](#)
- record_run_statistics, [823](#)
- record_walltime, [823](#)
- solve_profile_bound, [823](#)
- total_backtracks, [827](#)
- total_der_evals, [827](#)
- total_fun_evals, [827](#)
- total_iterations, [827](#)
- total_walltime, [828](#)
- mappel::estimator::QuasiNewtonMaximizer< Model >, [807](#)
- mappel::estimator::SimulatedAnnealingMaximizer
 - clear_stats, [830](#)
 - compute_profile_bound, [830](#)
 - compute_profile_bound_debug, [831](#)
 - Default_T_Init, [838](#)
 - DefaultCoolingRate, [838](#)
 - DefaultNumIterations, [838](#)
 - estimate_max, [831](#), [832](#)
 - estimate_max_debug, [832](#), [833](#)
 - estimate_max_stack, [833](#), [834](#)
 - estimate_profile_bounds, [834](#)
 - estimate_profile_bounds_debug, [835](#)
 - estimate_profile_bounds_parallel, [835](#)
 - estimate_profile_bounds_stack, [835](#)
 - estimate_profile_max, [836](#)
 - exit_counts, [839](#)
 - get_debug_stats, [836](#)
 - get_exit_counts, [837](#)
 - get_model, [837](#)
 - get_stats, [837](#)
 - max_threads, [839](#)
 - model, [839](#)
 - mtx, [839](#)
 - name, [837](#)
 - num_estimations, [839](#)
 - num_threads, [840](#)
 - record_exit_code, [837](#)
 - record_walltime, [838](#)
 - SimulatedAnnealingMaximizer, [830](#)
 - total_walltime, [840](#)
- mappel::estimator::SimulatedAnnealingMaximizer< Model >, [828](#)
- mappel::estimator::ThreadedEstimator
 - clear_stats, [865](#)
 - compute_estimate, [865](#)
 - compute_estimate_debug, [866](#)
 - compute_profile_bound, [866](#)
 - compute_profile_bound_debug, [866](#)
 - compute_profile_estimate, [866](#)
 - estimate_max, [867](#), [868](#)
 - estimate_max_debug, [868](#), [869](#)
 - estimate_max_stack, [869](#), [870](#)
 - estimate_profile_bounds, [870](#)
 - estimate_profile_bounds_debug, [870](#)
 - estimate_profile_bounds_parallel, [871](#)
 - estimate_profile_bounds_stack, [871](#)
 - estimate_profile_max, [872](#)
 - exit_counts, [874](#)
 - get_debug_stats, [872](#)
 - get_exit_counts, [873](#)
 - get_model, [873](#)
 - get_stats, [873](#)
 - max_threads, [874](#)
 - model, [875](#)
 - mtx, [875](#)
 - name, [873](#)
 - num_estimations, [875](#)
 - num_threads, [875](#)
 - record_exit_code, [874](#)
 - record_walltime, [874](#)
 - ThreadedEstimator, [865](#)
 - total_walltime, [876](#)
- mappel::estimator::ThreadedEstimator< Model >, [863](#)
- mappel::estimator::TrustRegionMaximizer
 - backtrack, [880](#)
 - backtrack_max_ratio, [894](#)
 - backtrack_min_linear_step_ratio, [894](#)
 - backtrack_min_ratio, [894](#)
 - clear_stats, [880](#)
 - compute_estimate, [880](#)
 - compute_estimate_debug, [881](#)
 - compute_profile_bound, [881](#)
 - compute_profile_bound_debug, [881](#)
 - compute_profile_estimate, [882](#)
 - convergence_min_function_change_ratio, [894](#)
 - convergence_min_step_size_ratio, [894](#)
 - convergence_min_trust_radius, [895](#)
 - convergence_test_grad_ratio, [882](#)
 - convergence_test_step_size, [882](#)
 - DefaultIterations, [895](#)
 - estimate_max, [882](#), [883](#)
 - estimate_max_debug, [884](#)
 - estimate_max_stack, [885](#)
 - estimate_profile_bounds, [886](#)
 - estimate_profile_bounds_debug, [886](#)
 - estimate_profile_bounds_parallel, [887](#)

- estimate_profile_bounds_stack, 887
- estimate_profile_max, 888
- exit_counts, 895
- get_debug_stats, 888
- get_exit_counts, 889
- get_model, 889
- get_stats, 889
- get_total_backtracks, 889
- get_total_der_evals, 890
- get_total_fun_evals, 890
- get_total_iterations, 890
- last_backtrack_idx, 895
- local_maximize, 890
- local_profile_maximize, 890
- max_backtracks, 895
- max_iterations, 895
- max_threads, 896
- maximize, 891
- MaximizerData, 879
- mean_backtracks, 891
- mean_der_evals, 892
- mean_fun_evals, 892
- mean_iterations, 892
- min_eigenvalue_correction_delta, 896
- min_profile_bound_residual, 896
- model, 896
- mtx, 896
- name, 892
- num_estimations, 897
- num_threads, 897
- profile_bound_backtrack, 892
- record_exit_code, 892
- record_run_statistics, 893
- record_walltime, 893
- rho_cauchy_min, 897
- rho_obj_min, 897
- rho_obj_opt, 897
- solve_profile_bound, 893
- total_backtracks, 898
- total_der_evals, 898
- total_fun_evals, 898
- total_iterations, 898
- total_walltime, 898
- trust_radius_decrease, 898
- trust_radius_decrease_min, 899
- trust_radius_increase, 899
- TrustRegionMaximizer, 880
- mappel::estimator::TrustRegionMaximizer< Model >, 876
- mappel::estimator::subroutine, 50
 - bound_step, 51
 - compute_D_scale, 52
 - compute_bound_scaling_vec, 51
 - compute_cauchy_point, 52
 - compute_initial_trust_radius, 52
 - compute_quadratic_model_value, 52
 - compute_scaled_problem, 52
 - solve_TR_subproblem, 53
 - solve_profile_initial_step, 53
 - solve_restricted_step_length_newton, 53
- mappel::mcmc, 53
 - compute_posterior_credible, 54
 - estimate_sample_posterior, 54
 - num_oversample, 54
 - sample_posterior, 54
 - sample_posterior_debug, 54
 - thin_sample, 55
- mappel::mcmc::MCMCData, 696
 - confidence, 696
 - credible_lb, 696
 - credible_ub, 697
 - initialize_arrays, 696
 - Nburnin, 697
 - Nsample, 697
 - sample, 697
 - sample_cov, 697
 - sample_mean, 697
 - sample_rllh, 697
 - thin, 698
- mappel::mcmc::MCMCDataStack, 698
 - confidence, 699
 - credible_lb, 699
 - credible_ub, 699
 - initialize_arrays, 699
 - Nburnin, 699
 - Ndata, 699
 - Nsample, 699
 - sample, 699
 - sample_cov, 699
 - sample_mean, 700
 - sample_rllh, 700
 - thin, 700
- mappel::mcmc::MCMCDebugData, 700
 - candidate, 701
 - candidate_rllh, 701
 - initialize_arrays, 701
 - Nsample, 701
 - sample, 701
 - sample_rllh, 701
- mappel::methods, 55
 - aposteriori_objective, 58
 - cr_lower_bound, 59
 - error_bounds_expected, 59
 - error_bounds_observed, 59
 - error_bounds_posterior_credible, 59
 - error_bounds_profile_likelihood, 59, 60
 - estimate_max, 60
 - estimate_posterior, 60, 61
 - estimate_profile_likelihood, 61

- expected_information, [61](#), [62](#)
- likelihood_objective, [62](#)
- make_estimator, [63](#)
- model_image, [63](#)
- observed_information, [64](#)
- prior_objective, [64](#)
- simulate_image, [64](#), [65](#)
- simulate_image_from_model, [65](#), [66](#)
- mappel::methods::debug, [66](#)
 - error_bounds_profile_likelihood_debug, [67](#)
 - estimate_max_debug, [67](#)
 - estimate_posterior_debug, [67](#)
- mappel::methods::likelihood, [67](#)
 - grad, [68](#)
 - grad2, [68](#), [69](#)
 - hessian, [69](#)
 - llh, [69](#)
 - rllh, [69](#), [70](#)
- mappel::methods::likelihood::debug, [70](#)
 - grad_components, [71](#)
 - hessian_components, [71](#)
 - llh_components, [71](#)
 - rllh_components, [71](#), [72](#)
- mappel::methods::objective, [72](#)
 - grad, [73](#), [74](#)
 - grad2, [74](#)
 - hessian, [74](#), [75](#)
 - llh, [75](#)
 - negative_definite_hessian, [76](#)
 - rllh, [76](#)
- mappel::methods::objective::debug, [77](#)
 - grad_components, [78](#)
 - hessian_components, [78](#)
 - llh_components, [78](#), [79](#)
 - rllh_components, [79](#)
- mappel::methods::objective::openmp, [79](#)
 - grad_stack, [80](#)
 - hessian_stack, [80](#)
 - llh_stack, [81](#), [82](#)
 - negative_definite_hessian_stack, [82](#)
 - rllh_stack, [83](#)
- mappel::methods::openmp, [84](#)
 - cr_lower_bound_stack, [85](#)
 - error_bounds_expected_stack, [85](#)
 - error_bounds_observed_stack, [85](#)
 - error_bounds_profile_likelihood_parallel, [85](#)
 - error_bounds_profile_likelihood_stack, [86](#)
 - estimate_max_stack, [86](#)
 - estimate_posterior_stack, [87](#)
 - estimate_profile_likelihood_stack, [87](#)
 - expected_information_stack, [87](#)
 - model_image_stack, [87](#)
 - sample_prior_stack, [88](#)
 - simulate_image_stack, [88](#)
- MappelError
 - mappel, [27](#)
- MatT
 - mappel, [28](#)
- max_backtracks
 - mappel::estimator::IterativeMaximizer, [587](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [739](#)
 - mappel::estimator::NewtonMaximizer, [760](#)
 - mappel::estimator::QuasiNewtonMaximizer, [825](#)
 - mappel::estimator::TrustRegionMaximizer, [895](#)
- max_iterations
 - mappel::estimator::IterativeMaximizer, [587](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [739](#)
 - mappel::estimator::NewtonMaximizer, [760](#)
 - mappel::estimator::QuasiNewtonMaximizer, [825](#)
 - mappel::estimator::TrustRegionMaximizer, [895](#)
- max_seq_len
 - mappel::estimator::IterativeMaximizer::Maximizer←Data, [613](#)
- max_threads
 - mappel::estimator::CGaussHeuristicEstimator, [102](#)
 - mappel::estimator::CGaussMLE, [114](#)
 - mappel::estimator::HeuristicEstimator, [555](#)
 - mappel::estimator::IterativeMaximizer, [587](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [739](#)
 - mappel::estimator::NewtonMaximizer, [760](#)
 - mappel::estimator::QuasiNewtonMaximizer, [825](#)
 - mappel::estimator::SimulatedAnnealingMaximizer, [839](#)
 - mappel::estimator::ThreadedEstimator, [874](#)
 - mappel::estimator::TrustRegionMaximizer, [896](#)
- MaxBacktracks
 - mappel::estimator, [50](#)
- MaxIter
 - mappel::estimator, [50](#)
- maxidx
 - mappel, [40](#)
- maximize
 - mappel::estimator::IterativeMaximizer, [583](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [735](#)
 - mappel::estimator::NewtonMaximizer, [756](#)
 - mappel::estimator::QuasiNewtonMaximizer, [821](#)
 - mappel::estimator::TrustRegionMaximizer, [891](#)
- MaximizerData
 - mappel::estimator::IterativeMaximizer::Maximizer←Data, [608](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [724](#)
 - mappel::estimator::NewtonMaximizer, [745](#)
 - mappel::estimator::QuasiNewtonMaximizer, [810](#)
 - mappel::estimator::TrustRegionMaximizer, [879](#)
- mcmc.cpp, [928](#)
- mcmc.h, [929](#)
- mcmc_candidate_eta_sigma
 - mappel::Gauss2DsxyModel, [541](#)

- mcmc_candidate_eta_y
 - mappel::Gauss2DsxyModel, 541
- mcmc_data.h, 930
- mean_backtracks
 - mappel::estimator::IterativeMaximizer, 583
 - mappel::estimator::NewtonDiagonalMaximizer, 735
 - mappel::estimator::NewtonMaximizer, 756
 - mappel::estimator::QuasiNewtonMaximizer, 821
 - mappel::estimator::TrustRegionMaximizer, 891
- mean_der_evals
 - mappel::estimator::IterativeMaximizer, 584
 - mappel::estimator::NewtonDiagonalMaximizer, 736
 - mappel::estimator::NewtonMaximizer, 757
 - mappel::estimator::QuasiNewtonMaximizer, 822
 - mappel::estimator::TrustRegionMaximizer, 892
- mean_fun_evals
 - mappel::estimator::IterativeMaximizer, 584
 - mappel::estimator::NewtonDiagonalMaximizer, 736
 - mappel::estimator::NewtonMaximizer, 757
 - mappel::estimator::QuasiNewtonMaximizer, 822
 - mappel::estimator::TrustRegionMaximizer, 892
- mean_iterations
 - mappel::estimator::IterativeMaximizer, 584
 - mappel::estimator::NewtonDiagonalMaximizer, 736
 - mappel::estimator::NewtonMaximizer, 757
 - mappel::estimator::QuasiNewtonMaximizer, 822
 - mappel::estimator::TrustRegionMaximizer, 892
- min_eigenvalue_correction_delta
 - mappel::estimator::IterativeMaximizer, 587
 - mappel::estimator::NewtonDiagonalMaximizer, 740
 - mappel::estimator::NewtonMaximizer, 761
 - mappel::estimator::QuasiNewtonMaximizer, 826
 - mappel::estimator::TrustRegionMaximizer, 896
- min_profile_bound_residual
 - mappel::estimator::IterativeMaximizer, 587
 - mappel::estimator::NewtonDiagonalMaximizer, 740
 - mappel::estimator::NewtonMaximizer, 761
 - mappel::estimator::QuasiNewtonMaximizer, 826
 - mappel::estimator::TrustRegionMaximizer, 896
- min_sigma
 - mappel::Gauss2DsMAP, 423
 - mappel::Gauss2DsMLE, 456
 - mappel::Gauss2DsModel, 489
 - mappel::Gauss2DsxyModel, 542
- mle
 - mappel::estimator::ProfileBoundsData, 802
 - mappel::estimator::ProfileBoundsDataStack, 805
- model
 - mappel::Gauss1DModel::Stencil, 863
 - mappel::Gauss1DsModel::Stencil, 849
 - mappel::Gauss2DModel::Stencil, 853
 - mappel::Gauss2DsModel::Stencil, 859
 - mappel::Gauss2DsxyModel::Stencil, 844
 - mappel::estimator::CGaussHeuristicEstimator, 102
 - mappel::estimator::CGaussMLE, 115
 - mappel::estimator::Estimator, 126
 - mappel::estimator::HeuristicEstimator, 555
 - mappel::estimator::IterativeMaximizer, 587
 - mappel::estimator::NewtonDiagonalMaximizer, 740
 - mappel::estimator::NewtonMaximizer, 761
 - mappel::estimator::QuasiNewtonMaximizer, 826
 - mappel::estimator::SimulatedAnnealingMaximizer, 839
 - mappel::estimator::ThreadedEstimator, 875
 - mappel::estimator::TrustRegionMaximizer, 896
- model_grad
 - PoissonGaussianNoise2DObjective.h, 953
- model_grad2
 - PoissonGaussianNoise2DObjective.h, 954
- model_hessian
 - PoissonGaussianNoise2DObjective.h, 954
- model_image
 - mappel::methods, 63
 - PoissonGaussianNoise2DObjective.h, 954
- model_image_stack
 - mappel::methods::openmp, 87
- model_methods.h, 938
- model_methods_impl.h, 941
- ModelBoundsError
 - mappel::ModelBoundsError, 721
- ModelDataStackT
 - mappel, 28
 - mappel::Gauss1DMAP, 133
 - mappel::Gauss1DMLE, 162
 - mappel::Gauss1DsMAP, 219
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss2DMAP, 307
 - mappel::Gauss2DMLE, 338
 - mappel::Gauss2DsMAP, 400
 - mappel::Gauss2DsMLE, 433
 - mappel::Gauss2DsxyMAP, 497
 - mappel::PoissonNoise1DObjective, 789
 - mappel::PoissonNoise2DObjective, 796
 - PoissonGaussianNoise2DObjective, 785
- ModelDataT
 - mappel, 28
 - mappel::Gauss1DMAP, 133
 - mappel::Gauss1DMLE, 162
 - mappel::Gauss1DsMAP, 219
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss2DMAP, 307
 - mappel::Gauss2DMLE, 338
 - mappel::Gauss2DsMAP, 400
 - mappel::Gauss2DsMLE, 433
 - mappel::Gauss2DsxyMAP, 498
 - mappel::PoissonNoise1DObjective, 789
 - mappel::PoissonNoise2DObjective, 796
 - PoissonGaussianNoise2DObjective, 785

ModellImprovement
 mappel::estimator, [50](#)
 modified_cholesky
 mappel, [40](#)
 mtx
 mappel::estimator::CGaussHeuristicEstimator, [102](#)
 mappel::estimator::CGaussMLE, [115](#)
 mappel::estimator::HeuristicEstimator, [555](#)
 mappel::estimator::IterativeMaximizer, [588](#)
 mappel::estimator::NewtonDiagonalMaximizer, [740](#)
 mappel::estimator::NewtonMaximizer, [761](#)
 mappel::estimator::QuasiNewtonMaximizer, [826](#)
 mappel::estimator::SimulatedAnnealingMaximizer, [839](#)
 mappel::estimator::ThreadedEstimator, [875](#)
 mappel::estimator::TrustRegionMaximizer, [896](#)
 nBacktracks
 mappel::estimator::IterativeMaximizer::Maximizer↵
 Data, [613](#)
 nIterations
 mappel::estimator::IterativeMaximizer::Maximizer↵
 Data, [614](#)
 name
 mappel::Gauss1DMP, [153](#)
 mappel::Gauss1DMLE, [182](#)
 mappel::Gauss1DsMAP, [240](#)
 mappel::Gauss1DsMLE, [269](#)
 mappel::Gauss2DMP, [328](#)
 mappel::Gauss2DMLE, [359](#)
 mappel::Gauss2DsMAP, [423](#)
 mappel::Gauss2DsMLE, [457](#)
 mappel::Gauss2DsxyMAP, [517](#)
 mappel::estimator::CGaussHeuristicEstimator, [101](#)
 mappel::estimator::CGaussMLE, [113](#)
 mappel::estimator::Estimator, [125](#)
 mappel::estimator::HeuristicEstimator, [554](#)
 mappel::estimator::IterativeMaximizer, [584](#)
 mappel::estimator::NewtonDiagonalMaximizer, [736](#)
 mappel::estimator::NewtonMaximizer, [757](#)
 mappel::estimator::QuasiNewtonMaximizer, [822](#)
 mappel::estimator::SimulatedAnnealingMaximizer, [837](#)
 mappel::estimator::ThreadedEstimator, [873](#)
 mappel::estimator::TrustRegionMaximizer, [892](#)
 Nburnin
 mappel::mcmc::MCMCData, [697](#)
 mappel::mcmc::MCMCDataStack, [699](#)
 Ndata
 mappel::estimator::ProfileBoundsDataStack, [805](#)
 mappel::mcmc::MCMCDataStack, [699](#)
 negative_definite_hessian
 mappel::methods::objective, [76](#)
 negative_definite_hessian_stack
 mappel::methods::objective::openmp, [82](#)
 NewtonDiagonalMaximizer
 mappel::estimator::NewtonDiagonalMaximizer, [724](#)
 NewtonMaximizer
 mappel::estimator::NewtonMaximizer, [745](#)
 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, [764](#)
 Nparams_est
 mappel::estimator::ProfileBoundsData, [802](#)
 mappel::estimator::ProfileBoundsDataStack, [805](#)
 Nsample
 mappel::mcmc::MCMCData, [697](#)
 mappel::mcmc::MCMCDataStack, [699](#)
 mappel::mcmc::MCMCDebugData, [701](#)
 Nseq
 mappel::estimator::MLEDebugData, [702](#)
 num_dim
 mappel::Gauss1DMP, [154](#)
 mappel::Gauss1DMLE, [183](#)
 mappel::Gauss1DModel, [210](#)
 mappel::Gauss1DsMAP, [240](#)
 mappel::Gauss1DsMLE, [269](#)
 mappel::Gauss1DsModel, [298](#)
 mappel::Gauss2DMP, [329](#)
 mappel::Gauss2DMLE, [360](#)
 mappel::Gauss2DModel, [390](#)
 mappel::Gauss2DsMAP, [424](#)
 mappel::Gauss2DsMLE, [457](#)
 mappel::Gauss2DsModel, [489](#)
 mappel::Gauss2DsxyMAP, [517](#)
 mappel::Gauss2DsxyModel, [542](#)
 mappel::ImageFormat1DBase, [562](#)
 mappel::ImageFormat2DBase, [568](#)
 mappel::PoissonNoise1DObjective, [792](#)
 mappel::PoissonNoise2DObjective, [800](#)
 num_estimations
 mappel::estimator::CGaussHeuristicEstimator, [103](#)
 mappel::estimator::CGaussMLE, [115](#)
 mappel::estimator::Estimator, [127](#)
 mappel::estimator::HeuristicEstimator, [556](#)
 mappel::estimator::IterativeMaximizer, [588](#)
 mappel::estimator::NewtonDiagonalMaximizer, [740](#)
 mappel::estimator::NewtonMaximizer, [761](#)
 mappel::estimator::QuasiNewtonMaximizer, [826](#)

- mappel::estimator::SimulatedAnnealingMaximizer, 839
- mappel::estimator::ThreadedEstimator, 875
- mappel::estimator::TrustRegionMaximizer, 897
- num_fixed_parameters
 - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 610
- num_hyperparams
 - mappel::Gauss1DMap, 154
 - mappel::Gauss1DMLE, 183
 - mappel::Gauss1DModel, 211
 - mappel::Gauss1DsMAP, 240
 - mappel::Gauss1DsMLE, 269
 - mappel::Gauss1DsModel, 298
 - mappel::Gauss2DMap, 329
 - mappel::Gauss2DMLE, 360
 - mappel::Gauss2DModel, 390
 - mappel::Gauss2DsMAP, 424
 - mappel::Gauss2DsMLE, 457
 - mappel::Gauss2DsModel, 489
 - mappel::Gauss2DsxyMAP, 517
 - mappel::Gauss2DsxyModel, 542
 - mappel::MAPEstimator, 605
 - mappel::MCMCAdaptor1Ds, 652
 - mappel::MCMCAdaptor1D, 633
 - mappel::MCMCAdaptor2Ds, 691
 - mappel::MCMCAdaptor2D, 671
 - mappel::MLEstimator, 719
 - mappel::PointEmitterModel, 783
- num_oversample
 - mappel::mcmc, 54
- num_params
 - mappel::Gauss1DMap, 154
 - mappel::Gauss1DMLE, 183
 - mappel::Gauss1DModel, 211
 - mappel::Gauss1DsMAP, 240
 - mappel::Gauss1DsMLE, 270
 - mappel::Gauss1DsModel, 298
 - mappel::Gauss2DMap, 329
 - mappel::Gauss2DMLE, 360
 - mappel::Gauss2DModel, 390
 - mappel::Gauss2DsMAP, 424
 - mappel::Gauss2DsMLE, 457
 - mappel::Gauss2DsModel, 489
 - mappel::Gauss2DsxyMAP, 517
 - mappel::Gauss2DsxyModel, 542
 - mappel::MAPEstimator, 605
 - mappel::MCMCAdaptor1Ds, 652
 - mappel::MCMCAdaptor1D, 633
 - mappel::MCMCAdaptor2Ds, 691
 - mappel::MCMCAdaptor2D, 672
 - mappel::MLEstimator, 719
 - mappel::PointEmitterModel, 783
- mappel::estimator::IterativeMaximizer::Maximizer↔Data, 614
- num_phases
 - mappel::Gauss1DMap, 154
 - mappel::Gauss1DMLE, 183
 - mappel::Gauss1DModel, 211
 - mappel::Gauss1DsMAP, 241
 - mappel::Gauss1DsMLE, 270
 - mappel::Gauss1DsModel, 298
 - mappel::Gauss2DMap, 329
 - mappel::Gauss2DMLE, 360
 - mappel::Gauss2DModel, 390
 - mappel::Gauss2DsMAP, 424
 - mappel::Gauss2DsMLE, 457
 - mappel::Gauss2DsModel, 489
 - mappel::Gauss2DsxyMAP, 518
 - mappel::MCMCAdaptor1Ds, 652
 - mappel::MCMCAdaptor1D, 633
 - mappel::MCMCAdaptor2Ds, 692
 - mappel::MCMCAdaptor2D, 672
 - mappel::MCMCAdaptorBase, 695
- num_threads
 - mappel::estimator::CGaussHeuristicEstimator, 103
 - mappel::estimator::CGaussMLE, 115
 - mappel::estimator::HeuristicEstimator, 556
 - mappel::estimator::IterativeMaximizer, 588
 - mappel::estimator::NewtonDiagonalMaximizer, 741
 - mappel::estimator::NewtonMaximizer, 762
 - mappel::estimator::QuasiNewtonMaximizer, 827
 - mappel::estimator::SimulatedAnnealingMaximizer, 840
 - mappel::estimator::ThreadedEstimator, 875
 - mappel::estimator::TrustRegionMaximizer, 897
- numerical.cpp, 944
- numerical.h, 945
- NumericalError
 - mappel::NumericalError, 764
- OMPExcptionCatcher
 - omp_exception_catcher, 89
 - omp_exception_catcher::impl_::OMPExcption↔Catcher, 765
- OMPExcptionCatcher.h, 946
- observed_information
 - mappel::methods, 64
- obsI
 - mappel::estimator::MLEDebugData, 702
- omp_exception_catcher, 89
 - Abort, 90
 - Continue, 90
 - DoNotTry, 90
 - OMPExcptionCatcher, 89
 - RethrowFirst, 90
 - Strategy, 90

- omp_exception_catcher::impl_, 90
- omp_exception_catcher::impl_::OMPExceptionCatcher
 - OMPExceptionCatcher, 765
 - rethrow, 766
 - run, 766
 - setGlobalDefaultStrategy, 766
- omp_exception_catcher::impl_::OMPExceptionCatcher<
 - _dummy >, 765
- openmp_methods.h, 947
- operator<<
 - mappel, 41
 - mappel::Gauss1DModel::Stencil, 862
 - mappel::Gauss1DsModel::Stencil, 848
 - mappel::Gauss2DModel::Stencil, 852
 - mappel::Gauss2DsModel::Stencil, 857
 - mappel::Gauss2DsxyModel::Stencil, 843
 - mappel::estimator, 50
 - mappel::estimator::Estimator, 126
- operator=
 - mappel::Gauss1DMAP, 144
 - mappel::Gauss1DMLE, 173
 - mappel::Gauss1DModel, 201
 - mappel::Gauss1DsMAP, 229, 230
 - mappel::Gauss1DsMLE, 259
 - mappel::Gauss1DsModel, 288
 - mappel::Gauss2DMAP, 318, 319
 - mappel::Gauss2DMLE, 349, 350
 - mappel::Gauss2DModel, 380
 - mappel::Gauss2DsMAP, 412
 - mappel::Gauss2DsMLE, 446
 - mappel::Gauss2DsModel, 478
 - mappel::ImageFormat2DBase, 567, 568
 - mappel::MAPEstimator, 599, 600
 - mappel::MCMCAdaptor1Ds, 644
 - mappel::MCMCAdaptor1D, 625
 - mappel::MCMCAdaptor2Ds, 683
 - mappel::MCMCAdaptor2D, 663, 664
 - mappel::MLEstimator, 713, 714
 - mappel::PointEmitterModel, 777
 - mappel::PoissonNoise1DObjective, 791
 - mappel::PoissonNoise2DObjective, 799
- ParallelRngGeneratorT
 - mappel, 28
- ParallelRngManagerT
 - mappel, 28
- ParamVecT
 - mappel, 28
 - mappel::Gauss1DMAP, 133
 - mappel::Gauss1DMLE, 162
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DsMAP, 219
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 278
- mappel::Gauss2DMAP, 307
- mappel::Gauss2DMLE, 338
- mappel::Gauss2DModel, 369
- mappel::Gauss2DsMAP, 400
- mappel::Gauss2DsMLE, 434
- mappel::Gauss2DsModel, 467
- mappel::Gauss2DsxyMAP, 498
- mappel::Gauss2DsxyModel, 526
- mappel::MAPEstimator, 593
- mappel::MCMCAdaptor1Ds, 637
- mappel::MCMCAdaptor1D, 618
- mappel::MCMCAdaptor2Ds, 676
- mappel::MCMCAdaptor2D, 657
- mappel::MLEstimator, 707
- mappel::PointEmitterModel, 771
- ParameterValueError
 - mappel::ParameterValueError, 767
- ParamT
 - mappel, 28
 - mappel::Gauss1DMAP, 133
 - mappel::Gauss1DMLE, 162
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DModel::Stencil, 860
 - mappel::Gauss1DsMAP, 219
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 278
 - mappel::Gauss1DsModel::Stencil, 846
 - mappel::Gauss2DMAP, 307
 - mappel::Gauss2DMLE, 338
 - mappel::Gauss2DModel, 369
 - mappel::Gauss2DModel::Stencil, 850
 - mappel::Gauss2DsMAP, 400
 - mappel::Gauss2DsMLE, 433
 - mappel::Gauss2DsModel, 467
 - mappel::Gauss2DsModel::Stencil, 855
 - mappel::Gauss2DsxyMAP, 498
 - mappel::Gauss2DsxyModel, 525
 - mappel::Gauss2DsxyModel::Stencil, 841
 - mappel::MAPEstimator, 593
 - mappel::MCMCAdaptor1Ds, 637
 - mappel::MCMCAdaptor1D, 618
 - mappel::MCMCAdaptor2Ds, 676
 - mappel::MCMCAdaptor2D, 657
 - mappel::MLEstimator, 707
 - mappel::PointEmitterModel, 771
- pareto_prior_grad
 - mappel, 42
- pareto_prior_grad2
 - mappel, 42
- pixel_grad
 - mappel::Gauss1DMAP, 144
 - mappel::Gauss1DMLE, 173
 - mappel::Gauss1DModel, 201
 - mappel::Gauss1DsMAP, 230

- mappel::Gauss1DsMLE, [259](#)
- mappel::Gauss1DsModel, [288](#)
- mappel::Gauss2DMAP, [319](#)
- mappel::Gauss2DMLE, [350](#)
- mappel::Gauss2DModel, [380](#)
- mappel::Gauss2DsMAP, [413](#)
- mappel::Gauss2DsMLE, [446](#)
- mappel::Gauss2DsModel, [478](#)
- mappel::Gauss2DsxyMAP, [508](#)
- mappel::Gauss2DsxyModel, [535](#)
- pixel_grad2
 - mappel::Gauss1DMAP, [144](#)
 - mappel::Gauss1DMLE, [173](#)
 - mappel::Gauss1DModel, [201](#)
 - mappel::Gauss1DsMAP, [230](#)
 - mappel::Gauss1DsMLE, [259](#)
 - mappel::Gauss1DsModel, [288](#)
 - mappel::Gauss2DMAP, [319](#)
 - mappel::Gauss2DMLE, [350](#)
 - mappel::Gauss2DModel, [380](#)
 - mappel::Gauss2DsMAP, [413](#)
 - mappel::Gauss2DsMLE, [446](#)
 - mappel::Gauss2DsModel, [478](#)
 - mappel::Gauss2DsxyMAP, [508](#)
 - mappel::Gauss2DsxyModel, [535](#)
- pixel_hess
 - mappel::Gauss1DMAP, [144](#)
 - mappel::Gauss1DMLE, [173](#)
 - mappel::Gauss1DModel, [201](#)
 - mappel::Gauss1DsMAP, [230](#)
 - mappel::Gauss1DsMLE, [259](#)
 - mappel::Gauss1DsModel, [288](#)
 - mappel::Gauss2DMAP, [319](#)
 - mappel::Gauss2DMLE, [350](#)
 - mappel::Gauss2DModel, [381](#)
 - mappel::Gauss2DsMAP, [413](#)
 - mappel::Gauss2DsMLE, [446](#)
 - mappel::Gauss2DsModel, [479](#)
 - mappel::Gauss2DsxyMAP, [508](#)
 - mappel::Gauss2DsxyModel, [535](#)
- pixel_hess_update
 - mappel::Gauss1DMAP, [144](#)
 - mappel::Gauss1DMLE, [173](#)
 - mappel::Gauss1DModel, [201](#)
 - mappel::Gauss1DsMAP, [230](#)
 - mappel::Gauss1DsMLE, [260](#)
 - mappel::Gauss1DsModel, [288](#)
 - mappel::Gauss2DMAP, [319](#)
 - mappel::Gauss2DMLE, [350](#)
 - mappel::Gauss2DModel, [381](#)
 - mappel::Gauss2DsMAP, [413](#)
 - mappel::Gauss2DsMLE, [447](#)
 - mappel::Gauss2DsModel, [479](#)
 - mappel::Gauss2DsxyMAP, [508](#)
- mappel::Gauss2DsxyModel, [535](#)
- pixel_model_value
 - mappel::Gauss1DMAP, [145](#)
 - mappel::Gauss1DMLE, [174](#)
 - mappel::Gauss1DModel, [202](#)
 - mappel::Gauss1DsMAP, [230](#)
 - mappel::Gauss1DsMLE, [260](#)
 - mappel::Gauss1DsModel, [289](#)
 - mappel::Gauss2DMAP, [319](#)
 - mappel::Gauss2DMLE, [350](#)
 - mappel::Gauss2DModel, [381](#)
 - mappel::Gauss2DsMAP, [413](#)
 - mappel::Gauss2DsMLE, [447](#)
 - mappel::Gauss2DsModel, [479](#)
 - mappel::Gauss2DsxyMAP, [508](#)
 - mappel::Gauss2DsxyModel, [535](#)
- PointEmitterModel
 - mappel::PointEmitterModel, [771](#)
- PointEmitterModel.cpp, [950](#)
- PointEmitterModel.h, [951](#)
- poisson_log_likelihood
 - mappel, [42](#)
- PoissonGaussianNoise2DObjective
 - CoordIdxT, [785](#)
 - CoordStackT, [785](#)
 - CoordT, [785](#)
 - estimator_names, [785](#)
 - ModelDataStackT, [785](#)
 - ModelDataT, [785](#)
 - PoissonGaussianNoise2DObjective, [785](#)
 - sensor_bg_map, [785](#)
 - sensor_gain_map, [786](#)
- PoissonGaussianNoise2DObjective< ModelBase >, [784](#)
- PoissonGaussianNoise2DObjective.cpp, [952](#)
- PoissonGaussianNoise2DObjective.h, [952](#)
 - fisher_information, [953](#)
 - log_likelihood, [953](#)
 - make_estimator, [953](#)
 - model_grad, [953](#)
 - model_grad2, [954](#)
 - model_hessian, [954](#)
 - model_image, [954](#)
 - relative_log_likelihood, [954](#)
 - simulate_image, [954](#)
- PoissonNoise1DObjective
 - mappel::PoissonNoise1DObjective, [789](#), [790](#)
- PoissonNoise1DObjective.cpp, [955](#)
- PoissonNoise1DObjective.h, [955](#)
- PoissonNoise2DObjective
 - mappel::PoissonNoise2DObjective, [797](#)
- PoissonNoise2DObjective.cpp, [957](#)
- PoissonNoise2DObjective.h, [957](#)
- 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, 183
 - mappel::Gauss1DModel, 211
 - mappel::Gauss1DsMAP, 241
 - mappel::Gauss1DsMLE, 270
 - mappel::Gauss1DsModel, 299
 - mappel::Gauss2DMAP, 329
 - mappel::Gauss2DMLE, 360
 - mappel::Gauss2DModel, 391
 - mappel::Gauss2DsMAP, 424
 - mappel::Gauss2DsMLE, 458
 - mappel::Gauss2DsModel, 490
 - mappel::Gauss2DsxyMAP, 518
 - mappel::Gauss2DsxyModel, 542
 - mappel::MAPEstimator, 606
 - mappel::MCMCA adaptor1Ds, 653
 - mappel::MCMCA adaptor1D, 633
 - mappel::MCMCA adaptor2Ds, 692
 - mappel::MCMCA adaptor2D, 672
 - mappel::MLEstimator, 720
 - mappel::PointEmitterModel, 783
- prior_objective
 - mappel::methods, 64
- prior_types
 - mappel::Gauss1DMAP, 155
 - mappel::Gauss1DMLE, 184
 - mappel::Gauss1DModel, 211
 - mappel::Gauss1DsMAP, 241
 - mappel::Gauss1DsMLE, 270
 - mappel::Gauss1DsModel, 299
 - mappel::Gauss2DMAP, 330
 - mappel::Gauss2DMLE, 361
 - mappel::Gauss2DModel, 391
 - mappel::Gauss2DsMAP, 425
 - mappel::Gauss2DsMLE, 458
 - mappel::Gauss2DsModel, 490
 - mappel::Gauss2DsxyMAP, 518
- profile_bound_backtrack
 - mappel::estimator::IterativeMaximizer, 584
 - mappel::estimator::NewtonDiagonalMaximizer, 736
 - mappel::estimator::NewtonMaximizer, 757
 - mappel::estimator::QuasiNewtonMaximizer, 822
 - mappel::estimator::TrustRegionMaximizer, 892
- profile_lb
 - mappel::estimator::ProfileBoundsData, 802
- mappel::estimator::ProfileBoundsDataStack, 806
- profile_points_lb
 - mappel::estimator::ProfileBoundsData, 802
 - mappel::estimator::ProfileBoundsDataStack, 806
- profile_points_lb_rllh
 - mappel::estimator::ProfileBoundsData, 803
 - mappel::estimator::ProfileBoundsDataStack, 806
- profile_points_ub
 - mappel::estimator::ProfileBoundsData, 803
 - mappel::estimator::ProfileBoundsDataStack, 806
- profile_points_ub_rllh
 - mappel::estimator::ProfileBoundsData, 803
 - mappel::estimator::ProfileBoundsDataStack, 806
- profile_ub
 - mappel::estimator::ProfileBoundsData, 803
 - mappel::estimator::ProfileBoundsDataStack, 806
- psf_sigma
 - mappel::Gauss1DMAP, 155
 - mappel::Gauss1DMLE, 184
 - mappel::Gauss1DModel, 212
 - mappel::Gauss2DMAP, 330
 - mappel::Gauss2DMLE, 361
 - mappel::Gauss2DModel, 391
 - mappel::Gauss2DsxyMAP, 518
- QuasiNewtonMaximizer
 - mappel::estimator::QuasiNewtonMaximizer, 810
- README.md, 959
- record_backtrack
 - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 610
- record_exit_code
 - mappel::estimator::CGaussHeuristicEstimator, 101
 - mappel::estimator::CGaussMLE, 113
 - mappel::estimator::Estimator, 125
 - mappel::estimator::HeuristicEstimator, 554
 - mappel::estimator::IterativeMaximizer, 584
 - mappel::estimator::NewtonDiagonalMaximizer, 736
 - mappel::estimator::NewtonMaximizer, 757
 - mappel::estimator::QuasiNewtonMaximizer, 822
 - mappel::estimator::SimulatedAnnealingMaximizer, 837
 - mappel::estimator::ThreadedEstimator, 874
 - mappel::estimator::TrustRegionMaximizer, 892
- record_iteration
 - mappel::estimator::IterativeMaximizer::Maximizer↔Data, 610
- record_run_statistics
 - mappel::estimator::IterativeMaximizer, 584
 - mappel::estimator::NewtonDiagonalMaximizer, 737
 - mappel::estimator::NewtonMaximizer, 758
 - mappel::estimator::QuasiNewtonMaximizer, 823
 - mappel::estimator::TrustRegionMaximizer, 893
- record_walltime

- mappel::estimator::CGaussHeuristicEstimator, [101](#)
- mappel::estimator::CGaussMLE, [114](#)
- mappel::estimator::Estimator, [126](#)
- mappel::estimator::HeuristicEstimator, [554](#)
- mappel::estimator::IterativeMaximizer, [585](#)
- mappel::estimator::NewtonDiagonalMaximizer, [737](#)
- mappel::estimator::NewtonMaximizer, [758](#)
- mappel::estimator::QuasiNewtonMaximizer, [823](#)
- mappel::estimator::SimulatedAnnealingMaximizer, [838](#)
- mappel::estimator::ThreadedEstimator, [874](#)
- mappel::estimator::TrustRegionMaximizer, [893](#)
- refine_gaussian_2Dmax
 - mappel, [43](#)
- refine_gaussian_3Dmax
 - mappel, [43](#)
- reflected_theta
 - mappel::Gauss1DMAP, [145](#)
 - mappel::Gauss1DMLE, [174](#)
 - mappel::Gauss1DModel, [202](#)
 - mappel::Gauss1DsMAP, [231](#)
 - mappel::Gauss1DsMLE, [260](#)
 - mappel::Gauss1DsModel, [289](#)
 - mappel::Gauss2DMAP, [320](#)
 - mappel::Gauss2DMLE, [351](#)
 - mappel::Gauss2DModel, [381](#)
 - mappel::Gauss2DsMAP, [414](#)
 - mappel::Gauss2DsMLE, [447](#)
 - mappel::Gauss2DsModel, [479](#)
 - mappel::Gauss2DsxyMAP, [508](#)
 - mappel::Gauss2DsxyModel, [536](#)
 - mappel::MAPEstimator, [600](#)
 - mappel::MCMCAdaptor1Ds, [644](#)
 - mappel::MCMCAdaptor1D, [625](#)
 - mappel::MCMCAdaptor2Ds, [683](#)
 - mappel::MCMCAdaptor2D, [664](#)
 - mappel::MLEstimator, [714](#)
 - mappel::PointEmitterModel, [777](#)
- reflected_theta_stack
 - mappel::Gauss1DMAP, [145](#)
 - mappel::Gauss1DMLE, [174](#)
 - mappel::Gauss1DModel, [202](#)
 - mappel::Gauss1DsMAP, [231](#)
 - mappel::Gauss1DsMLE, [260](#)
 - mappel::Gauss1DsModel, [289](#)
 - mappel::Gauss2DMAP, [320](#)
 - mappel::Gauss2DMLE, [351](#)
 - mappel::Gauss2DModel, [381](#)
 - mappel::Gauss2DsMAP, [414](#)
 - mappel::Gauss2DsMLE, [447](#)
 - mappel::Gauss2DsModel, [479](#)
 - mappel::Gauss2DsxyMAP, [509](#)
 - mappel::Gauss2DsxyModel, [536](#)
 - mappel::MAPEstimator, [600](#)
- mappel::MCMCAdaptor1Ds, [645](#)
- mappel::MCMCAdaptor1D, [626](#)
- mappel::MCMCAdaptor2Ds, [684](#)
- mappel::MCMCAdaptor2D, [664](#)
- mappel::MLEstimator, [714](#)
- mappel::PointEmitterModel, [778](#)
- relative_log_likelihood
 - PoissonGaussianNoise2DObjective.h, [954](#)
- relative_poisson_log_likelihood
 - mappel, [44](#)
- rename_hyperparam
 - mappel::Gauss1DMAP, [145](#)
 - mappel::Gauss1DMLE, [174](#)
 - mappel::Gauss1DModel, [202](#)
 - mappel::Gauss1DsMAP, [231](#)
 - mappel::Gauss1DsMLE, [260](#)
 - mappel::Gauss1DsModel, [289](#)
 - mappel::Gauss2DMAP, [320](#)
 - mappel::Gauss2DMLE, [351](#)
 - mappel::Gauss2DModel, [381](#)
 - mappel::Gauss2DsMAP, [414](#)
 - mappel::Gauss2DsMLE, [447](#)
 - mappel::Gauss2DsModel, [480](#)
 - mappel::Gauss2DsxyMAP, [509](#)
 - mappel::Gauss2DsxyModel, [536](#)
 - mappel::MAPEstimator, [600](#)
 - mappel::MCMCAdaptor1Ds, [645](#)
 - mappel::MCMCAdaptor1D, [626](#)
 - mappel::MCMCAdaptor2Ds, [684](#)
 - mappel::MCMCAdaptor2D, [664](#)
 - mappel::MLEstimator, [714](#)
 - mappel::PointEmitterModel, [778](#)
- restore_stencil
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [611](#)
- restrict_value_range
 - mappel, [44](#)
- rethrow
 - omp_exception_catcher::impl_::OMPEXception↔
Catcher, [766](#)
- RethrowFirst
 - omp_exception_catcher, [90](#)
- ReturnIfSubclassT
 - mappel, [28](#)
- rho_cauchy_min
 - mappel::estimator::TrustRegionMaximizer, [897](#)
- rho_obj_min
 - mappel::estimator::TrustRegionMaximizer, [897](#)
- rho_obj_opt
 - mappel::estimator::TrustRegionMaximizer, [897](#)
- rlh
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [614](#)
 - mappel::estimator::MLEDebugData, [703](#)

- mappel::methods::likelihood, [69](#), [70](#)
 - mappel::methods::objective, [76](#)
- rllh_beta2_prior
 - mappel, [44](#)
- rllh_beta_prior
 - mappel, [44](#)
- rllh_components
 - mappel::methods::likelihood::debug, [71](#), [72](#)
 - mappel::methods::objective::debug, [79](#)
- rllh_gamma_prior
 - mappel, [44](#)
- rllh_normal_prior
 - mappel, [44](#)
- rllh_pareto_prior
 - mappel, [44](#)
- rllh_stack
 - mappel::methods::objective::openmp, [83](#)
- rng.cpp, [959](#)
- rng.h, [960](#)
- rng_manager
 - mappel, [45](#)
- RngSeedT
 - mappel, [28](#)
- run
 - omp_exception_catcher::impl_::OMPException←
Catcher, [766](#)
- s0
 - mappel::estimator::IterativeMaximizer::Maximizer←
Data, [614](#)
- s1
 - mappel::estimator::IterativeMaximizer::Maximizer←
Data, [614](#)
- sample
 - mappel::mcmc::MCMCData, [697](#)
 - mappel::mcmc::MCMCDataStack, [699](#)
 - mappel::mcmc::MCMCDebugData, [701](#)
- sample_cov
 - mappel::mcmc::MCMCData, [697](#)
 - mappel::mcmc::MCMCDataStack, [699](#)
- sample_mcmc_candidate
 - mappel::Gauss1DModel, [145](#)
 - mappel::Gauss1DMLE, [174](#)
 - mappel::Gauss1DModel, [202](#)
 - mappel::Gauss1DsMAP, [231](#)
 - mappel::Gauss1DsMLE, [260](#), [261](#)
 - mappel::Gauss1DsModel, [289](#)
 - mappel::Gauss2DModel, [320](#)
 - mappel::Gauss2DMLE, [351](#)
 - mappel::Gauss2DModel, [382](#)
 - mappel::Gauss2DsMAP, [414](#)
 - mappel::Gauss2DsMLE, [448](#)
 - mappel::Gauss2DsModel, [480](#)
 - mappel::Gauss2DsxyMAP, [509](#)
- mappel::Gauss2DsxyModel, [536](#)
 - mappel::MCMCAdaptor1Ds, [645](#)
 - mappel::MCMCAdaptor1D, [626](#)
 - mappel::MCMCAdaptor2Ds, [684](#)
 - mappel::MCMCAdaptor2D, [664](#)
- sample_mean
 - mappel::mcmc::MCMCData, [697](#)
 - mappel::mcmc::MCMCDataStack, [700](#)
- sample_posterior
 - mappel::mcmc, [54](#)
- sample_posterior_debug
 - mappel::mcmc, [54](#)
- sample_prior
 - mappel::Gauss1DModel, [146](#)
 - mappel::Gauss1DMLE, [175](#)
 - mappel::Gauss1DModel, [203](#)
 - mappel::Gauss1DsMAP, [231](#), [232](#)
 - mappel::Gauss1DsMLE, [261](#)
 - mappel::Gauss1DsModel, [290](#)
 - mappel::Gauss2DModel, [320](#), [321](#)
 - mappel::Gauss2DMLE, [351](#), [352](#)
 - mappel::Gauss2DModel, [382](#)
 - mappel::Gauss2DsMAP, [414](#), [415](#)
 - mappel::Gauss2DsMLE, [448](#)
 - mappel::Gauss2DsModel, [480](#)
 - mappel::Gauss2DsxyMAP, [509](#)
 - mappel::Gauss2DsxyModel, [536](#)
 - mappel::MAPEstimator, [600](#)
 - mappel::MCMCAdaptor1Ds, [645](#)
 - mappel::MCMCAdaptor1D, [626](#)
 - mappel::MCMCAdaptor2Ds, [684](#)
 - mappel::MCMCAdaptor2D, [665](#)
 - mappel::MLEstimator, [714](#)
 - mappel::PointEmitterModel, [778](#)
- sample_prior_stack
 - mappel::methods::openmp, [88](#)
- sample_rllh
 - mappel::mcmc::MCMCData, [697](#)
 - mappel::mcmc::MCMCDataStack, [700](#)
 - mappel::mcmc::MCMCDebugData, [701](#)
- save_stencil
 - mappel::estimator::IterativeMaximizer::Maximizer←
Data, [611](#)
- saved_stencil
 - mappel::estimator::IterativeMaximizer::Maximizer←
Data, [611](#)
- saved_theta
 - mappel::estimator::IterativeMaximizer::Maximizer←
Data, [611](#)
- sensor_bg_map
 - PoissonGaussianNoise2DObjective, [785](#)
- sensor_gain_map
 - PoissonGaussianNoise2DObjective, [786](#)
- seq_len

- mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [614](#)
- seq_rllh
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [615](#)
- sequence
 - mappel::estimator::MLEDebugData, [703](#)
- sequence_rllh
 - mappel::estimator::MLEDebugData, [703](#)
- set_background_mcmc_sampling
 - mappel::Gauss1DMap, [146](#)
 - mappel::Gauss1DMLE, [175](#)
 - mappel::Gauss1DModel, [203](#)
 - mappel::Gauss1DsMap, [232](#)
 - mappel::Gauss1DsMLE, [261](#)
 - mappel::Gauss1DsModel, [290](#)
 - mappel::Gauss2DMap, [321](#)
 - mappel::Gauss2DMLE, [352](#)
 - mappel::Gauss2DModel, [382](#)
 - mappel::Gauss2DsMap, [415](#)
 - mappel::Gauss2DsMLE, [448](#)
 - mappel::Gauss2DsModel, [481](#)
 - mappel::Gauss2DsxyMap, [510](#)
 - mappel::Gauss2DsxyModel, [536](#)
 - mappel::MAPEstimator, [601](#)
 - mappel::MCMCAdaptor1Ds, [646](#)
 - mappel::MCMCAdaptor1D, [627](#)
 - mappel::MCMCAdaptor2Ds, [685](#)
 - mappel::MCMCAdaptor2D, [665](#)
- set_bounds
 - mappel::Gauss1DMap, [146](#)
 - mappel::Gauss1DMLE, [175](#)
 - mappel::Gauss1DModel, [203](#)
 - mappel::Gauss1DsMap, [232](#)
 - mappel::Gauss1DsMLE, [261](#)
 - mappel::Gauss1DsModel, [290](#)
 - mappel::Gauss2DMap, [321](#)
 - mappel::Gauss2DMLE, [352](#)
 - mappel::Gauss2DModel, [382](#)
 - mappel::Gauss2DsMap, [415](#)
 - mappel::Gauss2DsMLE, [448](#)
 - mappel::Gauss2DsModel, [481](#)
 - mappel::Gauss2DsxyMap, [510](#)
 - mappel::Gauss2DsxyModel, [536](#)
 - mappel::MAPEstimator, [600](#)
 - mappel::MCMCAdaptor1Ds, [646](#)
 - mappel::MCMCAdaptor1D, [627](#)
 - mappel::MCMCAdaptor2Ds, [685](#)
 - mappel::MCMCAdaptor2D, [665](#)
 - mappel::MLEstimator, [714](#)
 - mappel::PointEmitterModel, [778](#)
- set_fixed_parameters
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [611](#)
- set_hyperparam_names
 - mappel::Gauss1DMap, [146](#)
- mappel::Gauss1DMLE, [175](#)
 - mappel::Gauss1DModel, [203](#)
 - mappel::Gauss1DsMap, [232](#)
 - mappel::Gauss1DsMLE, [261](#)
 - mappel::Gauss1DsModel, [290](#)
 - mappel::Gauss2DMap, [321](#)
 - mappel::Gauss2DMLE, [352](#)
 - mappel::Gauss2DModel, [383](#)
 - mappel::Gauss2DsMap, [415](#)
 - mappel::Gauss2DsMLE, [449](#)
 - mappel::Gauss2DsModel, [481](#)
 - mappel::Gauss2DsxyMap, [510](#)
 - mappel::Gauss2DsxyModel, [536](#)
 - mappel::MAPEstimator, [601](#)
 - mappel::MCMCAdaptor1Ds, [646](#)
 - mappel::MCMCAdaptor1D, [627](#)
 - mappel::MCMCAdaptor2Ds, [685](#)
 - mappel::MCMCAdaptor2D, [665](#)
 - mappel::MLEstimator, [715](#)
 - mappel::PointEmitterModel, [778](#)
- set_hyperparam_value
 - mappel::Gauss1DMap, [146](#)
 - mappel::Gauss1DMLE, [175](#)
 - mappel::Gauss1DModel, [203](#)
 - mappel::Gauss1DsMap, [232](#)
 - mappel::Gauss1DsMLE, [262](#)
 - mappel::Gauss1DsModel, [290](#)
 - mappel::Gauss2DMap, [321](#)
 - mappel::Gauss2DMLE, [352](#)
 - mappel::Gauss2DModel, [383](#)
 - mappel::Gauss2DsMap, [415](#)
 - mappel::Gauss2DsMLE, [449](#)
 - mappel::Gauss2DsModel, [481](#)
 - mappel::Gauss2DsxyMap, [510](#)
 - mappel::Gauss2DsxyModel, [537](#)
 - mappel::MAPEstimator, [601](#)
 - mappel::MCMCAdaptor1Ds, [646](#)
 - mappel::MCMCAdaptor1D, [627](#)
 - mappel::MCMCAdaptor2Ds, [685](#)
 - mappel::MCMCAdaptor2D, [665](#)
 - mappel::MLEstimator, [715](#)
 - mappel::PointEmitterModel, [778](#)
- set_hyperparams
 - mappel::Gauss1DMap, [147](#)
 - mappel::Gauss1DMLE, [176](#)
 - mappel::Gauss1DModel, [204](#)
 - mappel::Gauss1DsMap, [232](#)
 - mappel::Gauss1DsMLE, [262](#)
 - mappel::Gauss1DsModel, [291](#)
 - mappel::Gauss2DMap, [321](#)
 - mappel::Gauss2DMLE, [352](#)
 - mappel::Gauss2DModel, [383](#)
 - mappel::Gauss2DsMap, [415](#)
 - mappel::Gauss2DsMLE, [449](#)

- mappel::Gauss2DsModel, [481](#)
- mappel::Gauss2DsxyMAP, [510](#)
- mappel::Gauss2DsxyModel, [537](#)
- mappel::MAPEstimator, [601](#)
- mappel::MCMCAdaptor1Ds, [646](#)
- mappel::MCMCAdaptor1D, [627](#)
- mappel::MCMCAdaptor2Ds, [685](#)
- mappel::MCMCAdaptor2D, [666](#)
- mappel::MLEstimator, [715](#)
- mappel::PointEmitterModel, [779](#)
- set_image_in_stack
 - mappel::Gauss1DMAP, [147](#)
 - mappel::Gauss1DMLE, [176](#)
 - mappel::Gauss1DModel, [204](#)
 - mappel::Gauss1DsMAP, [233](#)
 - mappel::Gauss1DsMLE, [262](#)
 - mappel::Gauss1DsModel, [291](#)
 - mappel::Gauss2DMAP, [322](#)
 - mappel::Gauss2DMLE, [353](#)
 - mappel::Gauss2DModel, [383](#)
 - mappel::Gauss2DsMAP, [416](#)
 - mappel::Gauss2DsMLE, [449](#)
 - mappel::Gauss2DsModel, [481](#)
 - mappel::Gauss2DsxyMAP, [510](#)
 - mappel::Gauss2DsxyModel, [537](#)
 - mappel::ImageFormat1DBase, [561](#)
 - mappel::ImageFormat2DBase, [568](#)
 - mappel::PoissonNoise1DObjective, [792](#)
 - mappel::PoissonNoise2DObjective, [799](#)
- set_intensity_mcmc_sampling
 - mappel::Gauss1DMAP, [147](#)
 - mappel::Gauss1DMLE, [176](#)
 - mappel::Gauss1DModel, [204](#)
 - mappel::Gauss1DsMAP, [233](#)
 - mappel::Gauss1DsMLE, [262](#)
 - mappel::Gauss1DsModel, [291](#)
 - mappel::Gauss2DMAP, [322](#)
 - mappel::Gauss2DMLE, [353](#)
 - mappel::Gauss2DModel, [383](#)
 - mappel::Gauss2DsMAP, [416](#)
 - mappel::Gauss2DsMLE, [449](#)
 - mappel::Gauss2DsModel, [481](#)
 - mappel::Gauss2DsxyMAP, [511](#)
 - mappel::MCMCAdaptor1Ds, [646](#)
 - mappel::MCMCAdaptor1D, [627](#)
 - mappel::MCMCAdaptor2Ds, [685](#)
 - mappel::MCMCAdaptor2D, [666](#)
- set_lbound
 - mappel::Gauss1DMAP, [147](#)
 - mappel::Gauss1DMLE, [176](#)
 - mappel::Gauss1DModel, [204](#)
 - mappel::Gauss1DsMAP, [233](#)
 - mappel::Gauss1DsMLE, [262](#)
 - mappel::Gauss1DsModel, [291](#)
- mappel::Gauss2DMAP, [322](#)
- mappel::Gauss2DMLE, [353](#)
- mappel::Gauss2DModel, [383](#)
- mappel::Gauss2DsMAP, [416](#)
- mappel::Gauss2DsMLE, [449](#)
- mappel::Gauss2DsModel, [482](#)
- mappel::Gauss2DsxyMAP, [511](#)
- mappel::Gauss2DsxyModel, [537](#)
- mappel::MAPEstimator, [601](#)
- mappel::MCMCAdaptor1Ds, [647](#)
- mappel::MCMCAdaptor1D, [627](#)
- mappel::MCMCAdaptor2Ds, [686](#)
- mappel::MCMCAdaptor2D, [666](#)
- mappel::MLEstimator, [715](#)
- mappel::PointEmitterModel, [779](#)
- set_max_sigma
 - mappel::Gauss1DsMAP, [233](#)
 - mappel::Gauss1DsMLE, [262](#), [263](#)
 - mappel::Gauss1DsModel, [291](#)
 - mappel::Gauss2DsMAP, [416](#)
 - mappel::Gauss2DsMLE, [450](#)
 - mappel::Gauss2DsModel, [482](#)
 - mappel::Gauss2DsxyModel, [537](#)
- set_max_sigma_ratio
 - mappel::Gauss2DsMAP, [416](#)
 - mappel::Gauss2DsMLE, [450](#)
 - mappel::Gauss2DsModel, [482](#)
 - mappel::Gauss2DsxyModel, [537](#)
- set_mcmc_num_phases
 - mappel::Gauss1DMAP, [147](#)
 - mappel::Gauss1DMLE, [176](#)
 - mappel::Gauss1DModel, [204](#)
 - mappel::Gauss1DsMAP, [233](#)
 - mappel::Gauss1DsMLE, [263](#)
 - mappel::Gauss1DsModel, [292](#)
 - mappel::Gauss2DMAP, [322](#)
 - mappel::Gauss2DMLE, [353](#)
 - mappel::Gauss2DModel, [384](#)
 - mappel::Gauss2DsMAP, [416](#)
 - mappel::Gauss2DsMLE, [450](#)
 - mappel::Gauss2DsModel, [482](#)
 - mappel::Gauss2DsxyMAP, [511](#)
 - mappel::MCMCAdaptor1Ds, [647](#)
 - mappel::MCMCAdaptor1D, [628](#)
 - mappel::MCMCAdaptor2Ds, [686](#)
 - mappel::MCMCAdaptor2D, [666](#)
 - mappel::MCMCAdaptorBase, [694](#)
- set_mcmc_sigma_scale
 - mappel::Gauss1DMAP, [147](#)
 - mappel::Gauss1DMLE, [176](#)
 - mappel::Gauss1DModel, [204](#)
 - mappel::Gauss1DsMAP, [234](#)
 - mappel::Gauss1DsMLE, [263](#)
 - mappel::Gauss1DsModel, [292](#)

- mappel::Gauss2DModel, 384
- mappel::Gauss2DsMAP, 417
- mappel::Gauss2DsMLE, 450
- mappel::Gauss2DsModel, 482
- mappel::Gauss2DsxyMAP, 511
- mappel::MCMCAdaptor1Ds, 647
- mappel::MCMCAdaptor1D, 628
- mappel::MCMCAdaptor2Ds, 686
- mappel::MCMCAdaptor2D, 666
- mappel::MCMCAdaptorBase, 695
- set_min_sigma
 - mappel::Gauss1DsMAP, 234
 - mappel::Gauss1DsMLE, 263
 - mappel::Gauss1DsModel, 292
 - mappel::Gauss2DsMAP, 417
 - mappel::Gauss2DsMLE, 450
 - mappel::Gauss2DsModel, 482
 - mappel::Gauss2DsxyModel, 537
- set_param_names
 - mappel::Gauss1DModel, 205
 - mappel::Gauss1DsMAP, 234
 - mappel::Gauss1DsMLE, 263
 - mappel::Gauss1DsModel, 292
 - mappel::Gauss2DModel, 384
 - mappel::Gauss2DsMAP, 417
 - mappel::Gauss2DsMLE, 450
 - mappel::Gauss2DsModel, 483
 - mappel::Gauss2DsxyMAP, 511
 - mappel::Gauss2DsxyModel, 537
 - mappel::MAPEstimator, 601
 - mappel::MCMCAdaptor1Ds, 647
 - mappel::MCMCAdaptor1D, 628
 - mappel::MCMCAdaptor2Ds, 686
 - mappel::MCMCAdaptor2D, 666
 - mappel::MLEstimator, 715
 - mappel::PointEmitterModel, 779
- set_prior
 - mappel::Gauss1DModel, 205
 - mappel::Gauss1DsMAP, 234
 - mappel::Gauss1DsMLE, 264
 - mappel::Gauss1DsModel, 292
 - mappel::Gauss2DModel, 322, 323
 - mappel::Gauss2DModel, 353, 354
 - mappel::Gauss2DsMAP, 417
 - mappel::Gauss2DsMLE, 451
- mappel::Gauss2DsModel, 483
- mappel::Gauss2DsxyMAP, 511
- mappel::Gauss2DsxyModel, 537
- mappel::MAPEstimator, 601, 602
- mappel::MCMCAdaptor1Ds, 647
- mappel::MCMCAdaptor1D, 628
- mappel::MCMCAdaptor2Ds, 686
- mappel::MCMCAdaptor2D, 667
- mappel::MLEstimator, 715, 716
- mappel::PointEmitterModel, 779
- set_psf_sigma
 - mappel::Gauss1DModel, 205
 - mappel::Gauss2DModel, 384
 - mappel::Gauss2DsxyMAP, 512
- set_rng_seed
 - mappel::Gauss1DModel, 205
 - mappel::Gauss1DsMAP, 234
 - mappel::Gauss1DsMLE, 264
 - mappel::Gauss1DsModel, 293
 - mappel::Gauss2DModel, 385
 - mappel::Gauss2DsMAP, 417
 - mappel::Gauss2DsMLE, 451
 - mappel::Gauss2DsModel, 483
 - mappel::Gauss2DsxyMAP, 512
 - mappel::Gauss2DsxyModel, 537
 - mappel::MAPEstimator, 602
 - mappel::MCMCAdaptor1Ds, 648
 - mappel::MCMCAdaptor1D, 628
 - mappel::MCMCAdaptor2Ds, 687
 - mappel::MCMCAdaptor2D, 667
 - mappel::MLEstimator, 716
 - mappel::PointEmitterModel, 779
- set_size
 - mappel::Gauss1DModel, 205, 206
 - mappel::Gauss1DsMAP, 235
 - mappel::Gauss1DsMLE, 264
 - mappel::Gauss1DsModel, 293
 - mappel::Gauss2DModel, 323
 - mappel::Gauss2DModel, 354
 - mappel::Gauss2DModel, 385
 - mappel::Gauss2DsMAP, 417
 - mappel::Gauss2DsMLE, 451
 - mappel::Gauss2DsModel, 483
 - mappel::Gauss2DsxyMAP, 512

- mappel::Gauss2DsxyModel, [538](#)
- mappel::ImageFormat1DBase, [561](#), [562](#)
- mappel::ImageFormat2DBase, [568](#)
- mappel::PoissonNoise1DObjective, [792](#)
- mappel::PoissonNoise2DObjective, [799](#)
- set_stencil
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [612](#)
- set_ubound
 - mappel::Gauss1DMap, [149](#)
 - mappel::Gauss1DMLE, [178](#)
 - mappel::Gauss1DModel, [206](#)
 - mappel::Gauss1DsMap, [235](#)
 - mappel::Gauss1DsMLE, [264](#)
 - mappel::Gauss1DsModel, [293](#)
 - mappel::Gauss2DMap, [323](#)
 - mappel::Gauss2DMLE, [354](#)
 - mappel::Gauss2DModel, [385](#)
 - mappel::Gauss2DsMap, [418](#)
 - mappel::Gauss2DsMLE, [451](#)
 - mappel::Gauss2DsModel, [483](#)
 - mappel::Gauss2DsxyMap, [512](#)
 - mappel::Gauss2DsxyModel, [538](#)
 - mappel::MAPEstimator, [602](#)
 - mappel::MCMCAdaptor1Ds, [648](#)
 - mappel::MCMCAdaptor1D, [629](#)
 - mappel::MCMCAdaptor2Ds, [687](#)
 - mappel::MCMCAdaptor2D, [667](#)
 - mappel::MLEstimator, [716](#)
 - mappel::PointEmitterModel, [779](#)
- setGlobalDefaultStrategy
 - omp_exception_catcher::impl_::OMPEXception↔
Catcher, [766](#)
- sgn
 - mappel, [44](#)
- sigma
 - mappel::Gauss1DsModel::Stencil, [847](#)
- sigma_ratio
 - mappel::Gauss2DsModel::Stencil, [856](#)
- sigma_scale
 - mappel::Gauss1DMap, [155](#)
 - mappel::Gauss1DMLE, [184](#)
 - mappel::Gauss1DModel, [212](#)
 - mappel::Gauss1DsMap, [241](#)
 - mappel::Gauss1DsMLE, [271](#)
 - mappel::Gauss1DsModel, [299](#)
 - mappel::Gauss2DMap, [330](#)
 - mappel::Gauss2DMLE, [361](#)
 - mappel::Gauss2DModel, [391](#)
 - mappel::Gauss2DsMap, [425](#)
 - mappel::Gauss2DsMLE, [458](#)
 - mappel::Gauss2DsModel, [490](#)
 - mappel::Gauss2DsxyMap, [519](#)
 - mappel::MCMCAdaptor1Ds, [653](#)
 - mappel::MCMCAdaptor1D, [633](#)
 - mappel::MCMCAdaptor2Ds, [692](#)
 - mappel::MCMCAdaptor2D, [672](#)
 - mappel::MCMCAdaptorBase, [695](#)
- sigmaX
 - mappel::Gauss2DsModel::Stencil, [856](#)
 - mappel::Gauss2DsxyModel::Stencil, [842](#)
- sigmaY
 - mappel::Gauss2DsModel::Stencil, [856](#)
 - mappel::Gauss2DsxyModel::Stencil, [842](#)
- simulate_image
 - mappel::methods, [64](#), [65](#)
 - PoissonGaussianNoise2DObjective.h, [954](#)
- simulate_image_from_model
 - mappel::methods, [65](#), [66](#)
- simulate_image_stack
 - mappel::methods::openmp, [88](#)
- SimulatedAnnealingMaximizer
 - mappel::estimator::SimulatedAnnealingMaximizer,
[830](#)
- size
 - mappel::Gauss1DMap, [155](#)
 - mappel::Gauss1DMLE, [184](#)
 - mappel::Gauss1DModel, [212](#)
 - mappel::Gauss1DsMap, [242](#)
 - mappel::Gauss1DsMLE, [271](#)
 - mappel::Gauss1DsModel, [299](#)
 - mappel::Gauss2DMap, [330](#)
 - mappel::Gauss2DMLE, [361](#)
 - mappel::Gauss2DModel, [392](#)
 - mappel::Gauss2DsMap, [425](#)
 - mappel::Gauss2DsMLE, [458](#)
 - mappel::Gauss2DsModel, [490](#)
 - mappel::Gauss2DsxyMap, [519](#)
 - mappel::Gauss2DsxyModel, [543](#)
 - mappel::ImageFormat1DBase, [562](#)
 - mappel::ImageFormat2DBase, [569](#)
 - mappel::PoissonNoise1DObjective, [793](#)
 - mappel::PoissonNoise2DObjective, [800](#)
- solve_TR_subproblem
 - mappel::estimator::subroutine, [53](#)
- solve_profile_bound
 - mappel::estimator::IterativeMaximizer, [585](#)
 - mappel::estimator::NewtonDiagonalMaximizer, [737](#)
 - mappel::estimator::NewtonMaximizer, [758](#)
 - mappel::estimator::QuasiNewtonMaximizer, [823](#)
 - mappel::estimator::TrustRegionMaximizer, [893](#)
- 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, [861](#)
 - mappel::Gauss1DsModel::Stencil, [846](#)
 - mappel::Gauss2DModel::Stencil, [850](#)
 - mappel::Gauss2DsModel::Stencil, [855](#)
 - mappel::Gauss2DsxyModel::Stencil, [842](#)
- stencil
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [612](#)
- stencil.cpp, [961](#)
- stencil.h, [962](#)
- StencilVecT
 - mappel, [29](#)
 - mappel::Gauss1DMap, [133](#)
 - mappel::Gauss1DMLE, [162](#)
 - mappel::Gauss1DModel, [191](#)
 - mappel::Gauss1DsMap, [219](#)
 - mappel::Gauss1DsMLE, [248](#)
 - mappel::Gauss1DsModel, [278](#)
 - mappel::Gauss2DMap, [307](#)
 - mappel::Gauss2DMLE, [338](#)
 - mappel::Gauss2DModel, [369](#)
 - mappel::Gauss2DsMap, [400](#)
 - mappel::Gauss2DsMLE, [434](#)
 - mappel::Gauss2DsModel, [467](#)
 - mappel::Gauss2DsxyMap, [498](#)
 - mappel::Gauss2DsxyModel, [526](#)
- StencilT
 - mappel, [28](#)
- step
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [615](#)
- StepSize
 - mappel::estimator, [50](#)
- Strategy
 - omp_exception_catcher, [90](#)
- 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, [803](#)
 - mappel::estimator::ProfileBoundsDataStack, [807](#)
- theta
 - mappel::Gauss1DModel::Stencil, [863](#)
 - mappel::Gauss1DsModel::Stencil, [849](#)
 - mappel::Gauss2DModel::Stencil, [853](#)
 - mappel::Gauss2DsModel::Stencil, [859](#)
 - mappel::Gauss2DsxyModel::Stencil, [845](#)
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, [612](#)
 - mappel::estimator::MLEDebugData, [703](#)
- theta_in_bounds
 - mappel::Gauss1DMap, [149](#)
 - mappel::Gauss1DMLE, [178](#)
 - mappel::Gauss1DModel, [206](#)
 - mappel::Gauss1DsMap, [235](#)
 - mappel::Gauss1DsMLE, [264](#)
 - mappel::Gauss1DsModel, [293](#)
 - mappel::Gauss2DMap, [323](#)
 - mappel::Gauss2DMLE, [354](#)
 - mappel::Gauss2DModel, [385](#)
 - mappel::Gauss2DsMap, [418](#)
 - mappel::Gauss2DsMLE, [451](#)
 - mappel::Gauss2DsModel, [483](#)
 - mappel::Gauss2DsxyMap, [512](#)
 - mappel::Gauss2DsxyModel, [538](#)
 - mappel::MAPEstimator, [602](#)
 - mappel::MCMCAdaptor1Ds, [648](#)
 - mappel::MCMCAdaptor1D, [629](#)
 - mappel::MCMCAdaptor2Ds, [687](#)
 - mappel::MCMCAdaptor2D, [667](#)
 - mappel::MLEstimator, [716](#)
 - mappel::PointEmitterModel, [780](#)

- theta_seq
 - mappel::estimator::IterativeMaximizer::Maximizer↔
Data, 615
- theta_stack_in_bounds
 - mappel::Gauss1DMAP, 149
 - mappel::Gauss1DMLE, 178
 - mappel::Gauss1DModel, 206
 - mappel::Gauss1DsMAP, 235
 - mappel::Gauss1DsMLE, 265
 - mappel::Gauss1DsModel, 293
 - mappel::Gauss2DMAP, 324
 - mappel::Gauss2DMLE, 355
 - mappel::Gauss2DModel, 385
 - mappel::Gauss2DsMAP, 418
 - mappel::Gauss2DsMLE, 451
 - mappel::Gauss2DsModel, 484
 - mappel::Gauss2DsxyMAP, 513
 - mappel::Gauss2DsxyModel, 538
 - mappel::MAPEstimator, 602
 - mappel::MCMCAdaptor1Ds, 648
 - mappel::MCMCAdaptor1D, 629
 - mappel::MCMCAdaptor2Ds, 687
 - mappel::MCMCAdaptor2D, 667
 - mappel::MLEstimator, 716
 - mappel::PointEmitterModel, 780
- thin
 - mappel::mcmc::MCMCData, 698
 - mappel::mcmc::MCMCDataStack, 700
- thin_sample
 - mappel::mcmc, 55
- ThreadedEstimator
 - mappel::estimator::ThreadedEstimator, 865
- total_backtracks
 - mappel::estimator::IterativeMaximizer, 588
 - mappel::estimator::NewtonDiagonalMaximizer, 741
 - mappel::estimator::NewtonMaximizer, 762
 - mappel::estimator::QuasiNewtonMaximizer, 827
 - mappel::estimator::TrustRegionMaximizer, 898
- total_der_evals
 - mappel::estimator::IterativeMaximizer, 589
 - mappel::estimator::NewtonDiagonalMaximizer, 741
 - mappel::estimator::NewtonMaximizer, 762
 - mappel::estimator::QuasiNewtonMaximizer, 827
 - mappel::estimator::TrustRegionMaximizer, 898
- total_fun_evals
 - mappel::estimator::IterativeMaximizer, 589
 - mappel::estimator::NewtonDiagonalMaximizer, 741
 - mappel::estimator::NewtonMaximizer, 762
 - mappel::estimator::QuasiNewtonMaximizer, 827
 - mappel::estimator::TrustRegionMaximizer, 898
- total_iterations
 - mappel::estimator::IterativeMaximizer, 589
 - mappel::estimator::NewtonDiagonalMaximizer, 741
 - mappel::estimator::NewtonMaximizer, 762
- mappel::estimator::QuasiNewtonMaximizer, 827
- mappel::estimator::TrustRegionMaximizer, 898
- total_walltime
 - mappel::estimator::CGaussHeuristicEstimator, 103
 - mappel::estimator::CGaussMLE, 116
 - mappel::estimator::Estimator, 127
 - mappel::estimator::HeuristicEstimator, 556
 - mappel::estimator::IterativeMaximizer, 589
 - mappel::estimator::NewtonDiagonalMaximizer, 742
 - mappel::estimator::NewtonMaximizer, 763
 - mappel::estimator::QuasiNewtonMaximizer, 828
 - mappel::estimator::SimulatedAnnealingMaximizer, 840
 - mappel::estimator::ThreadedEstimator, 876
 - mappel::estimator::TrustRegionMaximizer, 898
- trust_radius_decrease
 - mappel::estimator::TrustRegionMaximizer, 898
- trust_radius_decrease_min
 - mappel::estimator::TrustRegionMaximizer, 899
- trust_radius_increase
 - mappel::estimator::TrustRegionMaximizer, 899
- TrustRegionMaximizer
 - mappel::estimator::TrustRegionMaximizer, 880
- TrustRegionRadius
 - mappel::estimator, 50
- ubound
 - mappel::Gauss1DMAP, 156
 - mappel::Gauss1DMLE, 185
 - mappel::Gauss1DModel, 212
 - mappel::Gauss1DsMAP, 242
 - mappel::Gauss1DsMLE, 271
 - mappel::Gauss1DsModel, 300
 - mappel::Gauss2DMAP, 331
 - mappel::Gauss2DMLE, 362
 - mappel::Gauss2DModel, 392
 - mappel::Gauss2DsMAP, 426
 - mappel::Gauss2DsMLE, 459
 - mappel::Gauss2DsModel, 491
 - mappel::Gauss2DsxyMAP, 519
 - mappel::Gauss2DsxyModel, 543
 - mappel::MAPEstimator, 606
 - mappel::MCMCAdaptor1Ds, 653
 - mappel::MCMCAdaptor1D, 634
 - mappel::MCMCAdaptor2Ds, 692
 - mappel::MCMCAdaptor2D, 673
 - mappel::MLEstimator, 720
 - mappel::PointEmitterModel, 783
- Unassigned
 - mappel::estimator, 50
- UniformDistT
 - mappel, 29
- update_internal_1D_estimators
 - mappel::Gauss2DsxyModel, 538

- update_internal_1Dsum_estimators
 - mappel::Gauss2DMAP, [324](#)
 - mappel::Gauss2DMLE, [355](#)
 - mappel::Gauss2DModel, [385](#)
 - mappel::Gauss2DsMAP, [418](#)
 - mappel::Gauss2DsMLE, [452](#)
 - mappel::Gauss2DsModel, [484](#)
- util.cpp, [964](#)
- util.h, [964](#)
- VecFieldT
 - mappel, [29](#)
- VecT
 - mappel, [29](#)
- X
 - mappel::Gauss1DModel::Stencil, [863](#)
 - mappel::Gauss1DsModel::Stencil, [849](#)
 - mappel::Gauss2DModel::Stencil, [853](#)
 - mappel::Gauss2DsModel::Stencil, [859](#)
 - mappel::Gauss2DsxyModel::Stencil, [845](#)
- x
 - mappel::Gauss1DModel::Stencil, [861](#)
 - mappel::Gauss1DsModel::Stencil, [847](#)
 - mappel::Gauss2DModel::Stencil, [851](#)
 - mappel::Gauss2DsModel::Stencil, [856](#)
 - mappel::Gauss2DsxyModel::Stencil, [842](#)
- x_model
 - mappel::Gauss2DMAP, [331](#)
 - mappel::Gauss2DMLE, [362](#)
 - mappel::Gauss2DModel, [392](#)
 - mappel::Gauss2DsMAP, [426](#)
 - mappel::Gauss2DsMLE, [459](#)
 - mappel::Gauss2DsModel, [491](#)
 - mappel::Gauss2DsxyModel, [543](#)
- Y
 - mappel::Gauss2DModel::Stencil, [853](#)
 - mappel::Gauss2DsModel::Stencil, [859](#)
 - mappel::Gauss2DsxyModel::Stencil, [845](#)
- y
 - mappel::Gauss2DModel::Stencil, [851](#)
 - mappel::Gauss2DsModel::Stencil, [856](#)
 - mappel::Gauss2DsxyModel::Stencil, [843](#)
- y_model
 - mappel::Gauss2DMAP, [331](#)
 - mappel::Gauss2DMLE, [362](#)
 - mappel::Gauss2DModel, [393](#)
 - mappel::Gauss2DsMAP, [426](#)
 - mappel::Gauss2DsMLE, [459](#)
 - mappel::Gauss2DsModel, [491](#)
 - mappel::Gauss2DsxyModel, [544](#)