

Mappel

Generated by Doxygen 1.8.11

Contents

1	MAPPEL	2
2	Install	4
3	Namespace Index	7
3.1	Namespace List	7
4	Hierarchical Index	8
4.1	Class Hierarchy	8
5	Class Index	11
5.1	Class List	11
6	File Index	14
6.1	File List	14
7	Namespace Documentation	17
7.1	mappel Namespace Reference	17
7.1.1	Detailed Description	25
7.1.2	Typedef Documentation	25
7.1.3	Function Documentation	28
7.1.4	Variable Documentation	47
7.2	mappel::mcmc Namespace Reference	49
7.2.1	Function Documentation	49
7.3	mappel::methods Namespace Reference	50
7.3.1	Detailed Description	53
7.3.2	Function Documentation	53
7.4	mappel::methods::debug Namespace Reference	61
7.4.1	Function Documentation	62
7.5	mappel::methods::likelihood Namespace Reference	62

7.5.1	Function Documentation	63
7.6	mappel::methods::likelihood::debug Namespace Reference	65
7.6.1	Function Documentation	66
7.7	mappel::methods::objective Namespace Reference	67
7.7.1	Function Documentation	68
7.8	mappel::methods::objective::debug Namespace Reference	72
7.8.1	Function Documentation	73
7.9	mappel::methods::objective::openmp Namespace Reference	75
7.9.1	Function Documentation	75
7.10	mappel::methods::openmp Namespace Reference	79
7.10.1	Function Documentation	80
7.11	omp_exception_catcher Namespace Reference	85
7.11.1	Typedef Documentation	85
7.12	omp_exception_catcher::impl_ Namespace Reference	85
8	Class Documentation	85
8.1	mappel::ArrayShapeError Struct Reference	85
8.1.1	Detailed Description	86
8.1.2	Constructor & Destructor Documentation	86
8.2	mappel::ArraySizeError Struct Reference	86
8.2.1	Detailed Description	86
8.2.2	Constructor & Destructor Documentation	87
8.3	mappel::CGaussHeuristicEstimator< Model > Class Template Reference	87
8.3.1	Detailed Description	88
8.3.2	Member Enumeration Documentation	89
8.3.3	Constructor & Destructor Documentation	89
8.3.4	Member Function Documentation	89
8.3.5	Member Data Documentation	94

8.4	mappel::CGaussMLE< Model > Class Template Reference	95
8.4.1	Detailed Description	97
8.4.2	Member Enumeration Documentation	97
8.4.3	Constructor & Destructor Documentation	97
8.4.4	Member Function Documentation	98
8.4.5	Member Data Documentation	102
8.5	mappel::Estimator< Model > Class Template Reference	104
8.5.1	Detailed Description	105
8.5.2	Member Enumeration Documentation	105
8.5.3	Constructor & Destructor Documentation	106
8.5.4	Member Function Documentation	106
8.5.5	Friends And Related Function Documentation	110
8.5.6	Member Data Documentation	110
8.6	mappel::Gauss1DMAP Class Reference	111
8.6.1	Detailed Description	115
8.6.2	Member Typedef Documentation	116
8.6.3	Constructor & Destructor Documentation	118
8.6.4	Member Function Documentation	118
8.6.5	Member Data Documentation	133
8.7	mappel::Gauss1DMLE Class Reference	139
8.7.1	Detailed Description	143
8.7.2	Member Typedef Documentation	143
8.7.3	Constructor & Destructor Documentation	145
8.7.4	Member Function Documentation	146
8.7.5	Member Data Documentation	161
8.8	mappel::Gauss1DModel Class Reference	167
8.8.1	Detailed Description	171
8.8.2	Member Typedef Documentation	171

8.8.3	Constructor & Destructor Documentation	173
8.8.4	Member Function Documentation	173
8.8.5	Member Data Documentation	188
8.9	mappel::Gauss1DsMAP Class Reference	194
8.9.1	Detailed Description	198
8.9.2	Member Typedef Documentation	198
8.9.3	Constructor & Destructor Documentation	200
8.9.4	Member Function Documentation	201
8.9.5	Member Data Documentation	216
8.10	mappel::Gauss1DsMLE Class Reference	222
8.10.1	Detailed Description	226
8.10.2	Member Typedef Documentation	227
8.10.3	Constructor & Destructor Documentation	229
8.10.4	Member Function Documentation	229
8.10.5	Member Data Documentation	244
8.11	mappel::Gauss1DsModel Class Reference	250
8.11.1	Detailed Description	254
8.11.2	Member Typedef Documentation	254
8.11.3	Constructor & Destructor Documentation	256
8.11.4	Member Function Documentation	257
8.11.5	Member Data Documentation	272
8.12	mappel::Gauss2DMAP Class Reference	277
8.12.1	Detailed Description	282
8.12.2	Member Typedef Documentation	282
8.12.3	Constructor & Destructor Documentation	284
8.12.4	Member Function Documentation	285
8.12.5	Member Data Documentation	300
8.13	mappel::Gauss2DMLE Class Reference	307

8.13.1 Detailed Description	312
8.13.2 Member Typedef Documentation	312
8.13.3 Constructor & Destructor Documentation	314
8.13.4 Member Function Documentation	315
8.13.5 Member Data Documentation	330
8.14 mappel::Gauss2DModel Class Reference	337
8.14.1 Detailed Description	341
8.14.2 Member Typedef Documentation	341
8.14.3 Constructor & Destructor Documentation	343
8.14.4 Member Function Documentation	344
8.14.5 Member Data Documentation	359
8.15 mappel::Gauss2DsMAP Class Reference	366
8.15.1 Detailed Description	370
8.15.2 Member Typedef Documentation	370
8.15.3 Constructor & Destructor Documentation	373
8.15.4 Member Function Documentation	373
8.15.5 Member Data Documentation	391
8.16 mappel::Gauss2DsMLE Class Reference	398
8.16.1 Detailed Description	402
8.16.2 Member Typedef Documentation	402
8.16.3 Constructor & Destructor Documentation	405
8.16.4 Member Function Documentation	405
8.16.5 Member Data Documentation	423
8.17 mappel::Gauss2DsModel Class Reference	430
8.17.1 Detailed Description	435
8.17.2 Member Typedef Documentation	435
8.17.3 Constructor & Destructor Documentation	437
8.17.4 Member Function Documentation	437

8.17.5	Member Data Documentation	454
8.18	mappel::Gauss2DsxyMAP Class Reference	461
8.18.1	Detailed Description	465
8.18.2	Member Typedef Documentation	465
8.18.3	Constructor & Destructor Documentation	467
8.18.4	Member Function Documentation	467
8.18.5	Member Data Documentation	482
8.19	mappel::Gauss2DsxyModel Class Reference	488
8.19.1	Detailed Description	492
8.19.2	Member Typedef Documentation	492
8.19.3	Constructor & Destructor Documentation	494
8.19.4	Member Function Documentation	494
8.19.5	Member Data Documentation	506
8.20	mappel::HeuristicEstimator< Model > Class Template Reference	511
8.20.1	Detailed Description	513
8.20.2	Member Enumeration Documentation	513
8.20.3	Constructor & Destructor Documentation	513
8.20.4	Member Function Documentation	514
8.20.5	Member Data Documentation	518
8.21	mappel::ImageFormat1DBase Class Reference	519
8.21.1	Detailed Description	521
8.21.2	Member Typedef Documentation	521
8.21.3	Constructor & Destructor Documentation	522
8.21.4	Member Function Documentation	523
8.21.5	Member Data Documentation	525
8.22	mappel::ImageFormat2DBase Class Reference	526
8.22.1	Detailed Description	527
8.22.2	Member Typedef Documentation	527

8.22.3	Constructor & Destructor Documentation	529
8.22.4	Member Function Documentation	529
8.22.5	Member Data Documentation	531
8.23	mappel::IterativeMaximizer< Model > Class Template Reference	532
8.23.1	Detailed Description	534
8.23.2	Member Enumeration Documentation	535
8.23.3	Constructor & Destructor Documentation	535
8.23.4	Member Function Documentation	535
8.23.5	Member Data Documentation	541
8.24	mappel::LogicalError Struct Reference	544
8.24.1	Detailed Description	545
8.24.2	Constructor & Destructor Documentation	545
8.25	mappel::MAPEstimator Class Reference	545
8.25.1	Detailed Description	548
8.25.2	Member Typedef Documentation	548
8.25.3	Constructor & Destructor Documentation	548
8.25.4	Member Function Documentation	548
8.25.5	Member Data Documentation	557
8.26	mappel::IterativeMaximizer< Model >::MaximizerData Class Reference	560
8.26.1	Detailed Description	561
8.26.2	Constructor & Destructor Documentation	561
8.26.3	Member Function Documentation	562
8.26.4	Member Data Documentation	565
8.27	mappel::MCMCAdaptor1D Class Reference	568
8.27.1	Detailed Description	570
8.27.2	Member Typedef Documentation	570
8.27.3	Constructor & Destructor Documentation	571
8.27.4	Member Function Documentation	571

8.27.5	Member Data Documentation	581
8.28	mappel::MCMCAdaptor1Ds Class Reference	586
8.28.1	Detailed Description	588
8.28.2	Member Typedef Documentation	589
8.28.3	Constructor & Destructor Documentation	589
8.28.4	Member Function Documentation	589
8.28.5	Member Data Documentation	599
8.29	mappel::MCMCAdaptor2D Class Reference	604
8.29.1	Detailed Description	606
8.29.2	Member Typedef Documentation	607
8.29.3	Constructor & Destructor Documentation	607
8.29.4	Member Function Documentation	607
8.29.5	Member Data Documentation	617
8.30	mappel::MCMCAdaptor2Ds Class Reference	622
8.30.1	Detailed Description	625
8.30.2	Member Typedef Documentation	625
8.30.3	Constructor & Destructor Documentation	625
8.30.4	Member Function Documentation	626
8.30.5	Member Data Documentation	636
8.31	mappel::MCMCAdaptorBase Class Reference	641
8.31.1	Detailed Description	641
8.31.2	Constructor & Destructor Documentation	641
8.31.3	Member Function Documentation	642
8.31.4	Member Data Documentation	643
8.32	mappel::MLEstimator Class Reference	643
8.32.1	Detailed Description	646
8.32.2	Member Typedef Documentation	646
8.32.3	Constructor & Destructor Documentation	646

8.32.4	Member Function Documentation	646
8.32.5	Member Data Documentation	655
8.33	mappel::ModelBoundsError Struct Reference	658
8.33.1	Detailed Description	658
8.33.2	Constructor & Destructor Documentation	658
8.34	mappel::NewtonDiagonalMaximizer< Model > Class Template Reference	659
8.34.1	Detailed Description	661
8.34.2	Member Typedef Documentation	661
8.34.3	Member Enumeration Documentation	661
8.34.4	Constructor & Destructor Documentation	661
8.34.5	Member Function Documentation	662
8.34.6	Member Data Documentation	668
8.35	mappel::NewtonMaximizer< Model > Class Template Reference	672
8.35.1	Detailed Description	674
8.35.2	Member Typedef Documentation	674
8.35.3	Member Enumeration Documentation	674
8.35.4	Constructor & Destructor Documentation	674
8.35.5	Member Function Documentation	675
8.35.6	Member Data Documentation	681
8.36	mappel::NotImplementedError Struct Reference	685
8.36.1	Detailed Description	685
8.36.2	Constructor & Destructor Documentation	685
8.37	mappel::NumericalError Struct Reference	685
8.37.1	Detailed Description	686
8.37.2	Constructor & Destructor Documentation	686
8.38	omp_exception_catcher::impl_::OMPExcptionCatcher< IntType > Class Template Reference	686
8.38.1	Detailed Description	687
8.38.2	Member Enumeration Documentation	687

8.38.3	Constructor & Destructor Documentation	687
8.38.4	Member Function Documentation	687
8.39	mappel::ParameterValueError Struct Reference	688
8.39.1	Detailed Description	689
8.39.2	Constructor & Destructor Documentation	689
8.40	mappel::PointEmitterModel Class Reference	689
8.40.1	Detailed Description	692
8.40.2	Member Typedef Documentation	692
8.40.3	Constructor & Destructor Documentation	692
8.40.4	Member Function Documentation	693
8.40.5	Member Data Documentation	701
8.41	PoissonGaussianNoise2DObjective< ModelBase > Class Template Reference	704
8.41.1	Detailed Description	705
8.41.2	Member Typedef Documentation	705
8.41.3	Constructor & Destructor Documentation	706
8.41.4	Member Data Documentation	706
8.42	mappel::PoissonNoise1DObjective Class Reference	706
8.42.1	Detailed Description	708
8.42.2	Member Typedef Documentation	708
8.42.3	Constructor & Destructor Documentation	710
8.42.4	Member Function Documentation	710
8.42.5	Member Data Documentation	712
8.43	mappel::PoissonNoise2DObjective Class Reference	714
8.43.1	Detailed Description	715
8.43.2	Member Typedef Documentation	715
8.43.3	Constructor & Destructor Documentation	717
8.43.4	Member Function Documentation	717
8.43.5	Member Data Documentation	720

8.44	mappel::PriorMAP1DObjective Class Reference	. 721
8.44.1	Detailed Description	. 722
8.44.2	Member Typedef Documentation	. 722
8.44.3	Member Function Documentation	. 722
8.44.4	Member Data Documentation	. 723
8.45	mappel::QuasiNewtonMaximizer< Model > Class Template Reference	. 724
8.45.1	Detailed Description	. 726
8.45.2	Member Typedef Documentation	. 726
8.45.3	Member Enumeration Documentation	. 726
8.45.4	Constructor & Destructor Documentation	. 727
8.45.5	Member Function Documentation	. 727
8.45.6	Member Data Documentation	. 733
8.46	mappel::SimulatedAnnealingMaximizer< Model > Class Template Reference	. 737
8.46.1	Detailed Description	. 738
8.46.2	Member Enumeration Documentation	. 739
8.46.3	Constructor & Destructor Documentation	. 739
8.46.4	Member Function Documentation	. 739
8.46.5	Member Data Documentation	. 744
8.47	mappel::Gauss1DModel::Stencil Class Reference	. 746
8.47.1	Detailed Description	. 746
8.47.2	Member Typedef Documentation	. 746
8.47.3	Constructor & Destructor Documentation	. 747
8.47.4	Member Function Documentation	. 747
8.47.5	Friends And Related Function Documentation	. 748
8.47.6	Member Data Documentation	. 748
8.48	mappel::Gauss1DsModel::Stencil Class Reference	. 749
8.48.1	Detailed Description	. 750
8.48.2	Member Typedef Documentation	. 750

8.48.3	Constructor & Destructor Documentation	750
8.48.4	Member Function Documentation	751
8.48.5	Friends And Related Function Documentation	752
8.48.6	Member Data Documentation	752
8.49	mappel::Gauss2DsxyModel::Stencil Class Reference	753
8.49.1	Detailed Description	754
8.49.2	Member Typedef Documentation	754
8.49.3	Constructor & Destructor Documentation	755
8.49.4	Member Function Documentation	755
8.49.5	Friends And Related Function Documentation	756
8.49.6	Member Data Documentation	756
8.50	mappel::Gauss2DsModel::Stencil Class Reference	758
8.50.1	Detailed Description	759
8.50.2	Member Typedef Documentation	759
8.50.3	Constructor & Destructor Documentation	759
8.50.4	Member Function Documentation	760
8.50.5	Friends And Related Function Documentation	761
8.50.6	Member Data Documentation	761
8.51	mappel::Gauss2DModel::Stencil Class Reference	764
8.51.1	Detailed Description	765
8.51.2	Member Typedef Documentation	765
8.51.3	Constructor & Destructor Documentation	765
8.51.4	Member Function Documentation	766
8.51.5	Friends And Related Function Documentation	767
8.51.6	Member Data Documentation	767
8.52	mappel::ThreadedEstimator< Model > Class Template Reference	769
8.52.1	Detailed Description	770
8.52.2	Member Enumeration Documentation	771
8.52.3	Constructor & Destructor Documentation	771
8.52.4	Member Function Documentation	771
8.52.5	Member Data Documentation	776
8.53	mappel::TrustRegionMaximizer< Model > Class Template Reference	777
8.53.1	Detailed Description	780
8.53.2	Member Typedef Documentation	780
8.53.3	Member Enumeration Documentation	780
8.53.4	Constructor & Destructor Documentation	781
8.53.5	Member Function Documentation	781
8.53.6	Member Data Documentation	788

9 File Documentation	793
9.1 display.cpp File Reference	793
9.1.1 Detailed Description	794
9.2 display.h File Reference	794
9.2.1 Detailed Description	794
9.3 estimator.h File Reference	795
9.3.1 Detailed Description	795
9.3.2 Typedef Documentation	796
9.4 estimator_impl.h File Reference	796
9.4.1 Detailed Description	796
9.5 estimator_statics.cpp File Reference	796
9.5.1 Detailed Description	797
9.6 Gauss1DMAP.cpp File Reference	797
9.6.1 Detailed Description	797
9.7 Gauss1DMAP.h File Reference	797
9.7.1 Detailed Description	798
9.8 Gauss1DMLE.cpp File Reference	798
9.8.1 Detailed Description	798
9.9 Gauss1DMLE.h File Reference	799
9.9.1 Detailed Description	799
9.10 Gauss1DModel.cpp File Reference	799
9.10.1 Detailed Description	800
9.11 Gauss1DModel.h File Reference	800
9.11.1 Detailed Description	800
9.12 Gauss1DsMAP.cpp File Reference	801
9.12.1 Detailed Description	801
9.13 Gauss1DsMAP.h File Reference	801
9.13.1 Detailed Description	802

9.14 Gauss1DsMLE.cpp File Reference	802
9.14.1 Detailed Description	802
9.15 Gauss1DsMLE.h File Reference	802
9.15.1 Detailed Description	803
9.16 Gauss1DsModel.cpp File Reference	803
9.16.1 Detailed Description	803
9.17 Gauss1DsModel.h File Reference	804
9.17.1 Detailed Description	804
9.18 Gauss2DMP.cpp File Reference	804
9.18.1 Detailed Description	805
9.19 Gauss2DMP.h File Reference	805
9.19.1 Detailed Description	805
9.20 Gauss2DMLE.cpp File Reference	806
9.20.1 Detailed Description	806
9.21 Gauss2DMLE.h File Reference	806
9.21.1 Detailed Description	807
9.22 Gauss2DModel.cpp File Reference	807
9.22.1 Detailed Description	807
9.23 Gauss2DModel.h File Reference	808
9.23.1 Detailed Description	808
9.24 Gauss2DsMAP.cpp File Reference	809
9.24.1 Detailed Description	809
9.25 Gauss2DsMAP.h File Reference	809
9.25.1 Detailed Description	810
9.26 Gauss2DsMLE.cpp File Reference	810
9.26.1 Detailed Description	810
9.27 Gauss2DsMLE.h File Reference	810
9.27.1 Detailed Description	811

9.28 Gauss2DsModel.cpp File Reference	811
9.28.1 Detailed Description	811
9.29 Gauss2DsModel.h File Reference	812
9.29.1 Detailed Description	812
9.30 Gauss2DsxyMAP.h File Reference	813
9.30.1 Detailed Description	813
9.31 Gauss2DsxyModel.h File Reference	813
9.31.1 Detailed Description	814
9.32 ImageFormat1DBase.cpp File Reference	814
9.32.1 Detailed Description	815
9.33 ImageFormat1DBase.h File Reference	815
9.33.1 Detailed Description	815
9.34 ImageFormat2DBase.cpp File Reference	816
9.34.1 Detailed Description	816
9.35 ImageFormat2DBase.h File Reference	816
9.35.1 Detailed Description	817
9.36 Install.md File Reference	817
9.37 MAPEstimator.h File Reference	817
9.37.1 Detailed Description	818
9.38 mcmc.cpp File Reference	818
9.39 mcmc.h File Reference	819
9.39.1 Detailed Description	820
9.40 MCMCAdaptor1D.cpp File Reference	820
9.40.1 Detailed Description	820
9.41 MCMCAdaptor1D.h File Reference	820
9.41.1 Detailed Description	821
9.42 MCMCAdaptor1Ds.cpp File Reference	821
9.42.1 Detailed Description	821

9.43	MCMCAdaptor1Ds.h File Reference	822
9.43.1	Detailed Description	822
9.44	MCMCAdaptor2D.cpp File Reference	822
9.44.1	Detailed Description	822
9.45	MCMCAdaptor2D.h File Reference	823
9.45.1	Detailed Description	823
9.46	MCMCAdaptor2Ds.cpp File Reference	823
9.46.1	Detailed Description	823
9.47	MCMCAdaptor2Ds.h File Reference	824
9.47.1	Detailed Description	824
9.48	MCMCAdaptorBase.cpp File Reference	824
9.48.1	Detailed Description	824
9.49	MCMCAdaptorBase.h File Reference	825
9.49.1	Detailed Description	825
9.50	MLEstimator.h File Reference	825
9.50.1	Detailed Description	826
9.51	model_methods.h File Reference	827
9.52	model_methods_impl.h File Reference	830
9.53	numerical.cpp File Reference	833
9.53.1	Detailed Description	834
9.54	numerical.h File Reference	834
9.54.1	Detailed Description	835
9.55	OMPExcptionCatcher.h File Reference	835
9.55.1	Detailed Description	836
9.56	openmp_methods.h File Reference	836
9.56.1	Detailed Description	839
9.57	PointEmitterModel.cpp File Reference	839
9.57.1	Detailed Description	839

9.58	PointEmitterModel.h File Reference	840
9.58.1	Detailed Description	840
9.59	PoissonGaussianNoise2DObjective.cpp File Reference	841
9.60	PoissonGaussianNoise2DObjective.h File Reference	841
9.60.1	Detailed Description	842
9.60.2	Function Documentation	842
9.61	PoissonNoise1DObjective.cpp File Reference	844
9.61.1	Detailed Description	844
9.62	PoissonNoise1DObjective.h File Reference	845
9.62.1	Detailed Description	846
9.63	PoissonNoise2DObjective.cpp File Reference	846
9.63.1	Detailed Description	847
9.64	PoissonNoise2DObjective.h File Reference	847
9.64.1	Detailed Description	848
9.65	PriorMAP1DObjective.h File Reference	849
9.65.1	Detailed Description	849
9.66	README.md File Reference	849
9.67	rng.cpp File Reference	849
9.68	rng.h File Reference	850
9.68.1	Detailed Description	850
9.69	stackcomp.h File Reference	851
9.69.1	Detailed Description	852
9.70	stencil.cpp File Reference	852
9.70.1	Detailed Description	853
9.71	stencil.h File Reference	853
9.71.1	Detailed Description	854
9.72	util.cpp File Reference	855
9.73	util.h File Reference	855

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! ([[Apache-2.0](#)](LICENSE))

Documentation

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

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

Background

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

Figure 1: Effective fitting resolution in typical applications

Applications

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

Performance

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

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

Installation

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

Dependencies

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

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

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

External Projects

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

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

Model classes

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

Computations available

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

Design Notes

Static Polymorphism

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

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 Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

mappel	17
mappel::mcmc	49
mappel::methods	
Templated functions for operating on a PointEmitterModel	50

mappel::methods::debug	61
mappel::methods::likelihood	62
mappel::methods::likelihood::debug	65
mappel::methods::objective	67
mappel::methods::objective::debug	72
mappel::methods::objective::openmp	75
mappel::methods::openmp	79
omp_exception_catcher	85
omp_exception_catcher::impl_	85

4 Hierarchical Index

4.1 Class Hierarchy

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

mappel::Estimator< Model >	104
mappel::ThreadedEstimator< Model >	769
mappel::CGaussHeuristicEstimator< Model >	87
mappel::CGaussMLE< Model >	95
mappel::HeuristicEstimator< Model >	511
mappel::IterativeMaximizer< Model >	532
mappel::NewtonDiagonalMaximizer< Model >	659
mappel::NewtonMaximizer< Model >	672
mappel::QuasiNewtonMaximizer< Model >	724
mappel::TrustRegionMaximizer< Model >	777
mappel::SimulatedAnnealingMaximizer< Model >	737
mappel::ImageFormat1DBase	519
mappel::Gauss1DModel	167
mappel::Gauss1DMAP	111
mappel::Gauss1DMLE	139
mappel::Gauss2DsxyMAP	461

mappel::Gauss1DsModel	250
mappel::Gauss1DsMAP	194
mappel::Gauss1DsMLE	222
mappel::PoissonNoise1DObjective	706
mappel::Gauss1DMAP	111
mappel::Gauss1DMLE	139
mappel::Gauss1DsMAP	194
mappel::Gauss1DsMLE	222
mappel::Gauss2DsxyMAP	461
ImageFormat1DBase	
PoissonGaussianNoise2DObjective< ModelBase >	704
mappel::ImageFormat2DBase	526
mappel::Gauss2DModel	337
mappel::Gauss2DMAP	277
mappel::Gauss2DMLE	307
mappel::Gauss2DsModel	430
mappel::Gauss2DsMAP	366
mappel::Gauss2DsMLE	398
mappel::Gauss2DsxyModel	488
mappel::PoissonNoise2DObjective	714
mappel::Gauss2DMAP	277
mappel::Gauss2DMLE	307
mappel::Gauss2DsMAP	366
mappel::Gauss2DsMLE	398
MappelError	
mappel::ArrayShapeError	85
mappel::ArraySizeError	86
mappel::LogicalError	544
mappel::ModelBoundsError	658
mappel::NotImplementedError	685

mappel::NumericalError	685
mappel::ParameterValueError	688
mappel::IterativeMaximizer< Model >::MaximizerData	560
mappel::MCMCAdaptorBase	641
mappel::MCMCAdaptor1D	568
mappel::Gauss1DModel	167
mappel::MCMCAdaptor1Ds	586
mappel::Gauss1DsModel	250
mappel::MCMCAdaptor2D	604
mappel::Gauss2DModel	337
mappel::MCMCAdaptor2Ds	622
mappel::Gauss2DsModel	430
omp_exception_catcher::impl_::OMPExceptionCatcher< IntType >	686
mappel::PointEmitterModel	689
mappel::Gauss1DModel	167
mappel::Gauss1DsModel	250
mappel::Gauss2DModel	337
mappel::Gauss2DsModel	430
mappel::Gauss2DsxyModel	488
mappel::MAPEstimator	545
mappel::Gauss1DMAP	111
mappel::Gauss1DsMAP	194
mappel::Gauss2DMAP	277
mappel::Gauss2DsMAP	366
mappel::Gauss2DsxyMAP	461
mappel::MCMCAdaptor1D	568
mappel::MLEstimator	643
mappel::Gauss1DMLE	139
mappel::Gauss1DsMLE	222
mappel::Gauss2DMLE	307

mappel::Gauss2DsMLE	398
mappel::PriorMAP1DObjective	721
mappel::Gauss1DModel::Stencil	746
mappel::Gauss1DsModel::Stencil	749
mappel::Gauss2DsxyModel::Stencil	753
mappel::Gauss2DsModel::Stencil	758
mappel::Gauss2DModel::Stencil	764

5 Class Index

5.1 Class List

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

mappel::ArrayShapeError	
Array is not of the right dimensionality	85
mappel::ArraySizeError	
Array is not of the right size	86
mappel::CGaussHeuristicEstimator< Model >	87
mappel::CGaussMLE< Model >	95
mappel::Estimator< Model >	104
mappel::Gauss1DMAP	
A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	111
mappel::Gauss1DMLE	
A 1D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective	139
mappel::Gauss1DModel	
A base class for 1D Gaussian PSF with a fixed sigma (standard dev.)	167
mappel::Gauss1DsMAP	
A 1D Gaussian with variable PSF sigma under an Poisson read noise assumption and MAP Objective	194
mappel::Gauss1DsMLE	
A 1D Gaussian with variable PSF under an Poisson noise assumption and maximum-likelihood estimator	222
mappel::Gauss1DsModel	
Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels	250

<code>mappel::Gauss2DMap</code>	A 2D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	277
<code>mappel::Gauss2DMLE</code>	A 2D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective	307
<code>mappel::Gauss2DModel</code>	A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma	337
<code>mappel::Gauss2DsMAP</code>	A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum a-posteriori objective	366
<code>mappel::Gauss2DsMLE</code>	A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum-likelihood objective	398
<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>]	430
<code>mappel::Gauss2DsxyMAP</code>	A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	461
<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>]	488
<code>mappel::HeuristicEstimator< Model ></code>		511
<code>mappel::ImageFormat1DBase</code>	A virtual base class for 2D image localization objectives	519
<code>mappel::ImageFormat2DBase</code>	A virtual base class for 2D image localization objectives	526
<code>mappel::IterativeMaximizer< Model ></code>		532
<code>mappel::LogicalError</code>	Failure of code or algorithm logic	544
<code>mappel::MAPEstimator</code>	A Mixin class to configure a for MLE estimation (null prior)	545
<code>mappel::IterativeMaximizer< Model >::MaximizerData</code>		560
<code>mappel::MCMCAdaptor1D</code>		568
<code>mappel::MCMCAdaptor1Ds</code>		586
<code>mappel::MCMCAdaptor2D</code>		604
<code>mappel::MCMCAdaptor2Ds</code>		622

mappel::MCMCAdaptorBase	641
mappel::MLEstimator A Mixin class to configure a for MLE estimation (null prior)	643
mappel::ModelBoundsError Access outside the model bounds is attempted	658
mappel::NewtonDiagonalMaximizer< Model >	659
mappel::NewtonMaximizer< Model >	672
mappel::NotImplementedError Feature not yet implemented	685
mappel::NumericalError Expected numerical condition does not hold	685
omp_exception_catcher::impl_::OMPExcptionCatcher< IntType >	686
mappel::ParameterValueError Parameter value is not valid	688
mappel::PointEmitterModel A virtual Base type for point emitter localization models	689
PoissonGaussianNoise2DObjective< ModelBase > A Base type for point emitter localization models that use 2d images	704
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 phontons given a certain mean rate of incidence on each pixel	706
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 phontons given a certain mean rate of incidence on each pixel	714
mappel::PriorMAP1DObjective A Mixin class to configure a Gauss1DModel for MAP estimation (default 1D prior)	721
mappel::QuasiNewtonMaximizer< Model >	724
mappel::SimulatedAnnealingMaximizer< Model >	737
mappel::Gauss1DModel::Stencil Stencil for 1D fixed-sigma models	746
mappel::Gauss1DsModel::Stencil Stencil for 1D variable-sigma models	749
mappel::Gauss2DsxyModel::Stencil Stencil for 2D free-sigma (astigmatic) models	753
mappel::Gauss2DsModel::Stencil Stencil for 2D scalar-sigma models	758

mappel::Gauss2DModel::Stencil Stencil for 2D fixed-sigma models	764
mappel::ThreadedEstimator< Model >	769
mappel::TrustRegionMaximizer< Model >	777

6 File Index

6.1 File List

Here is a list of all files with brief descriptions:

display.cpp	793
display.h	794
estimator.h The class declaration and inline and templated functions for the Estimator class hierarchy	795
estimator_impl.h	796
estimator_statics.cpp	796
Gauss1DMAP.cpp The class definition and template Specializations for Gauss1DMAP	797
Gauss1DMAP.h The class declaration and inline and templated functions for Gauss1DMAP	797
Gauss1DMLE.cpp The class definition and template Specializations for Gauss1DMLE	798
Gauss1DMLE.h The class declaration and inline and templated functions for Gauss1DMLE	799
Gauss1DModel.cpp The class definition and template Specializations for Gauss1DModel	799
Gauss1DModel.h The class declaration and inline and templated functions for Gauss1DModel	800
Gauss1DsMAP.cpp The class definition and template Specializations for Gauss1DsMAP	801
Gauss1DsMAP.h The class declaration and inline and templated functions for Gauss1DsMAP	801
Gauss1DsMLE.cpp The class definition and template Specializations for Gauss1DsMLE	802
Gauss1DsMLE.h The class declaration and inline and templated functions for Gauss1DsMLE	802

Gauss1DsModel.cpp	803
The class definition and template Specializations for Gauss1DsModel	
Gauss1DsModel.h	804
The class declaration and inline and templated functions for Gauss1DsModel	
Gauss2DMAP.cpp	804
The class definition and template Specializations for Gauss2DMAP	
Gauss2DMAP.h	805
The class declaration and inline and templated functions for Gauss2DMAP	
Gauss2DMLE.cpp	806
The class definition and template Specializations for Gauss2DMLE	
Gauss2DMLE.h	806
The class declaration and inline and templated functions for Gauss2DMLE	
Gauss2DModel.cpp	807
The class definition and template Specializations for Gauss2DModel	
Gauss2DModel.h	808
The class declaration and inline and templated functions for Gauss2DModel	
Gauss2DsMAP.cpp	809
The class definition and template Specializations for Gauss2DsMAP	
Gauss2DsMAP.h	809
The class declaration and inline and templated functions for Gauss2DsMAP	
Gauss2DsMLE.cpp	810
The class definition and template Specializations for Gauss2DsMLE	
Gauss2DsMLE.h	810
The class declaration and inline and templated functions for Gauss2DsMLE	
Gauss2DsModel.cpp	811
The class definition and template Specializations for Gauss2DsModel	
Gauss2DsModel.h	812
The class declaration and inline and templated functions for Gauss2DsModel	
Gauss2DsxyMAP.h	813
The class declaration and inline and templated functions for Gauss2DsxyMAP	
Gauss2DsxyModel.h	813
The class declaration and inline and templated functions for Gauss2DsxyModel	
ImageFormat1DBase.cpp	814
The class definition and template Specializations for ImageFormat1DBase	
ImageFormat1DBase.h	815
The class declaration and inline and templated functions for ImageFormat1DBase	
ImageFormat2DBase.cpp	816
The class definition and template Specializations for ImageFormat2DBase	

ImageFormat2DBase.h	
The class declaration and inline and templated functions for ImageFormat2DBase	816
MAPEstimator.h	
Class declaration and inline and templated functions for MAPEstimator	817
mcmc.cpp	818
mcmc.h	
Templated MCMC methods for posterior estimation	819
MCMCAdaptor1D.cpp	
The class definition and template Specializations for MCMCAdaptor1D	820
MCMCAdaptor1D.h	
The class declaration and inline and templated functions for MCMCAdaptor1D	820
MCMCAdaptor1Ds.cpp	
The class definition and template Specializations for MCMCAdaptor1Ds	821
MCMCAdaptor1Ds.h	
The class declaration and inline and templated functions for MCMCAdaptor1Ds	822
MCMCAdaptor2D.cpp	
The class definition and template Specializations for MCMCAdaptor2D	822
MCMCAdaptor2D.h	
The class declaration and inline and templated functions for MCMCAdaptor2D	823
MCMCAdaptor2Ds.cpp	
The class definition and template Specializations for MCMCAdaptor2Ds	823
MCMCAdaptor2Ds.h	
The class declaration and inline and templated functions for MCMCAdaptor2Ds	824
MCMCAdaptorBase.cpp	
The class definition and template Specializations for MCMCAdaptorBase	824
MCMCAdaptorBase.h	
The class declaration and inline and templated functions for MCMCAdaptorBase	825
MLEstimator.h	
Class declaration and inline and templated functions for MLEstimator	825
model_methods.h	827
model_methods_impl.h	830
numerical.cpp	
Numerical matrix operations	833
numerical.h	
Numerical matrix operations	834
OMPExceptionCatcher.h	
A lightweight class for managing C++ exception handling strategies for openMP methods	835

openmp_methods.h	Namespaces for OpenMP parallelized versions of the mappel::model namespace functions (external methods)	836
PointEmitterModel.cpp	The class definition and template Specializations for PointEmitterModel	839
PointEmitterModel.h	The class declaration and inline and templated functions for PointEmitterModel	840
PoissonGaussianNoise2DObjective.cpp		841
PoissonGaussianNoise2DObjective.h	The class declaration and inline and templated functions for PoissonGaussianNoise2DObjective	841
PoissonNoise1DObjective.cpp	The class definition and template Specializations for PoissonNoise1DObjective	844
PoissonNoise1DObjective.h	The class declaration and inline and templated functions for PoissonNoise1DObjective	845
PoissonNoise2DObjective.cpp	The class definition and template Specializations for PoissonNoise2DObjective	846
PoissonNoise2DObjective.h	The class declaration and inline and templated functions for PoissonNoise2DObjective	847
PriorMAP1DObjective.h	The class declaration and inline and templated functions for PriorMAP1DObjective	849
rng.cpp		849
rng.h	Random number generation usign sfmt	850
stackcomp.h	Data-parallel versions of core computational functions using OpenMP	851
stencil.cpp	The stencils for pixel based computations	852
stencil.h	The stencils for pixel based computations	853
util.cpp		855
util.h		855

7 Namespace Documentation

7.1 mappel Namespace Reference

Namespaces

- [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 [CGaussHeuristicEstimator](#)

- class [CGaussMLE](#)

- class [Estimator](#)

- 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 [HeuristicEstimator](#)

- class [ImageFormat1DBase](#)

A virtual base class for 2D image localization objectives.

- class [ImageFormat2DBase](#)
A virtual base class for 2D image localization objectives.
- class [IterativeMaximizer](#)
- 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.
- class [NewtonDiagonalMaximizer](#)
- class [NewtonMaximizer](#)
- 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.
- class [PriorMAP1DObjective](#)
A Mixin class to configure a [Gauss1DModel](#) for MAP estimation (default 1D prior).
- class [QuasiNewtonMaximizer](#)
- class [SimulatedAnnealingMaximizer](#)
- class [ThreadedEstimator](#)
- class [TrustRegionMaximizer](#)

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_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 normal_quantile_twosided` (`double confidence`)
- `double normal_quantile_onesided` (`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 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::ostream & operator<<` (`std::ostream &out`, `Estimator< Model > &estimator`)
- `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 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)
Genrates 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)
- `template<class Model >`
`void sample_prior_stack` (Model &model, typename Model::ParamVecT &theta_stack)
Parallel sampling of the model prior.
- `template<class Model >`
`void model_image_stack` (const Model &model, const typename Model::ParamVecT &theta_stack, typename
Model::ImageStackT &image_stack)
Parallel computation of the model image.
- `template<class Model >`
`void simulate_image_stack` (const Model &model, const typename Model::ParamVecT &theta_stack, typename
Model::ImageStackT &image_stack)
Parallel simulation of images from one or more theta.
- `template<class Model >`
`void log_likelihood_stack` (const Model &model, const typename Model::ImageT &image, const typename
Model::ParamVecT &theta_stack, VecT &llh_stack)
Parallel log_likelihood calculations for a single image.

- `template<class Model >`
`void log_likelihood_stack (const Model &model, const typename Model::ImageStackT &image_stack, const type-`
`name Model::ParamVecT &theta_stack, VecT &llh_stack)`
Parallel log_likelihood calculations for a stack of images.
- `template<class Model >`
`void model_grad_stack (const Model &model, const typename Model::ImageStackT &image_stack, const type-`
`name Model::ParamVecT &theta_stack, typename Model::ParamVecT &grad_stack)`
Parallel model gradient calculations for a stack of images.
- `template<class Model >`
`void model_hessian_stack (const Model &model, const typename Model::ImageStackT &image_stack, const`
`typename Model::ParamVecT &theta_stack, CubeT &hessian_stack)`
Parallel model Hessian calculations for a stack of images.
- `template<class Model >`
`void model_positive_hessian_stack (const Model &model, const typename Model::ImageStackT &image_stack,`
`const typename Model::ParamVecT &theta_stack, CubeT &hessian_stack)`
Parallel model positive-definite Hessian approximation calculations for a stack of images.
- `template<class Model >`
`void cr_lower_bound_stack (const Model &model, const typename Model::ParamVecT &theta_stack, typename`
`Model::ParamVecT &crlb_stack)`
- `template<class Model >`
`void fisher_information_stack (const Model &model, const typename Model::ParamVecT &theta_stack, CubeT`
`&fisherl_stack)`
- `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

7.1.1 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.

7.1.2 Typedef Documentation

7.1.2.1 using `mappel::BoolT` = `typedef uint16_t`

Definition at line 19 of file `util.h`.

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

Definition at line 20 of file `util.h`.

7.1.2.3 using `mappel::CubeT` = `typedef arma::cube`

A type to represent floating-point data cubes

Definition at line 26 of file `util.h`.

7.1.2.4 `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 33 of file `util.h`.

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

A type to represent integer data arrays

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

7.1.2.6 using `mappel::IdxT` = `typedef arma::uword`

Definition at line 21 of file `util.h`.

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

A type to represent integer data arrays

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

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

Definition at line 38 of file util.h.

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

Definition at line 39 of file util.h.

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

Definition at line 47 of file util.h.

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

Definition at line 43 of file util.h.

7.1.2.12 `using mappel::MappelError = typedef backtrace_exception::BacktraceException`

Definition at line 60 of file util.h.

7.1.2.13 `using mappel::MatT = typedef arma::mat`

A type to represent floating-point data matrices

Definition at line 25 of file util.h.

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

Definition at line 48 of file util.h.

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

Definition at line 44 of file util.h.

7.1.2.16 `using mappel::ParallelRngGeneratorT = typedef trng::lcg64_shift`

Definition at line 21 of file rng.h.

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

Definition at line 22 of file rng.h.

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

Definition at line 41 of file util.h.

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

Definition at line 42 of file util.h.

7.1.2.20 `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 36 of file util.h.

7.1.2.21 `using mappel::RngSeedT = typedef parallel_rng::SeedT`

Definition at line 23 of file rng.h.

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

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

Definition at line 28 of file util.h.

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

Definition at line 45 of file util.h.

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

Definition at line 49 of file util.h.

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

Definition at line 29 of file util.h.

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

Definition at line 24 of file rng.h.

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

Definition at line 27 of file util.h.

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

A type to represent floating-point data arrays

Definition at line 24 of file util.h.

7.1.3 Function Documentation

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

Definition at line 314 of file stencil.h.

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

Definition at line 347 of file stencil.h.

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

Definition at line 307 of file stencil.h.

Referenced by `mappel::PriorMAP1DObjective::prior_grad_update()`.

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

Definition at line 339 of file stencil.h.

Referenced by `mappel::PriorMAP1DObjective::prior_grad2_update()`, and `mappel::PriorMAP1DObjective::prior_hess←_update()`.

7.1.3.5 `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::CGaussMLE< Model >::compute_estimate()`.

7.1.3.6 `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`.

7.1.3.7 `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`.

7.1.3.8 `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::CGaussMLE< Model >::compute_estimate_debug()`.

7.1.3.9 `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`.

7.1.3.10 `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`.

7.1.3.11 `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::CGaussMLE< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_↵
estimate_debug()`, and `mappel::HeuristicEstimator< Model >::get_debug_stats()`.

7.1.3.12 `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`.

7.1.3.13 `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`.

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

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

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

7.1.3.17 `bool mappel::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. If false then Usym is left in arbitrary corrupted state.

Definition at line 97 of file numerical.cpp.

Referenced by `is_positive_definite()`, `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`, and `mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem()`.

7.1.3.18 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()`, `cholesky_make_positive_definite()`, and `mappel::NewtonMaximizer< Model >::maximize()`.

7.1.3.19 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::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`.

7.1.3.20 void mappel::cholesky_make_negative_definite (arma::mat & m)

Modify m inplace using modified choslesky 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()`.

7.1.3.21 void mappel::cholesky_make_positive_definite (arma::mat & m)

Modify m inplace using modified choslesky decomposition to ensure m is positive definite

Definition at line 46 of file numerical.cpp.

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

7.1.3.22 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 184 of file numerical.cpp.

Referenced by `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::solve_↵_restricted_step_length_newton()`, and `mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem()`.

7.1.3.23 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 symmertic matrix, fill in the upper triangle by copying the lowerr triangle. This operation modifies the matrix.

Definition at line 30 of file numerical.cpp.

7.1.3.24 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 symmertic 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()`, and `mappel::NewtonMaximizer< Model >::maximize()`.

7.1.3.25 void mappel::copy_Usym_mat_stack (arma::cube & *usym_stack*)

Definition at line 20 of file numerical.cpp.

7.1.3.26 template<class Model > void mappel::cr_lower_bound_stack (const Model & *model*, const typename Model::ParamVecT & *theta_stack*, typename Model::ParamVecT & *crlb_stack*)

Definition at line 304 of file stackcomp.h.

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

7.1.3.27 void mappel::enable_all_cpus ()

Definition at line 10 of file util.cpp.

7.1.3.28 double mappel::estimate_background (const MatT & *im*, const MatT & *unit_model_im*, double *min_bg*)

Definition at line 263 of file stencil.cpp.

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

Definition at line 282 of file stencil.cpp.

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

Definition at line 141 of file stencil.cpp.

References `gaussian_convolution()`.

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

Definition at line 205 of file stencil.cpp.

References `gaussian_3D_convolution()`.

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

Definition at line 277 of file stencil.cpp.

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

Definition at line 292 of file stencil.cpp.

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

Definition at line 151 of file stencil.h.

Referenced by `make_d_stencil()`.

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

Definition at line 176 of file stencil.h.

Referenced by `make_DX_stencil()`.

7.1.3.36 `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 190 of file stencil.h.

Referenced by `make_DXS2_stencil()`.

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

Definition at line 183 of file stencil.h.

References `square()`.

Referenced by `make_DXS_stencil()`.

7.1.3.38 `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 203 of file stencil.h.

Referenced by `make_DXSX_stencil()`.

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

Definition at line 157 of file stencil.h.

References `square()`.

Referenced by `make_G_stencil()`.

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

Definition at line 40 of file stencil.cpp.

References `gauss_norm()`.

Referenced by `make_gaussian_stencil()`.

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

Definition at line 164 of file stencil.h.

Referenced by `make_X_stencil()`.

7.1.3.42 `template<class Model > void mappel::fisher_information_stack (const Model & model, const typename Model::ParamVecT & theta_stack, CubeT & fisherl_stack)`

Definition at line 315 of file stackcomp.h.

References `fisher_information()`.

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

Definition at line 320 of file stencil.h.

Referenced by `mappel::PriorMAP1DObjective::prior_grad_update()`.

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

Definition at line 356 of file stencil.h.

Referenced by `mappel::PriorMAP1DObjective::prior_grad2_update()`, and `mappel::PriorMAP1DObjective::prior_hess←_update()`.

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

Definition at line 92 of file stencil.h.

Referenced by `fill_gaussian_stencil()`.

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

Definition at line 184 of file stencil.cpp.

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

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

Definition at line 127 of file stencil.cpp.

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

7.1.3.48 `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()`.

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

Definition at line 57 of file rng.h.

Referenced by `generate_poisson()`.

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

Genrates a single poisson disributed 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. Trasformed to work ing log space. This is linear in mu. Works ok for small counts.

Definition at line 43 of file rng.h.

Referenced by generate_poisson().

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

Definition at line 39 of file util.cpp.

7.1.3.52 `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().

7.1.3.53 `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::QuasiNewtonMaximizer< Model >::maximize().

7.1.3.54 `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.

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

Definition at line 21 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()`.

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

Definition at line 29 of file util.cpp.

7.1.3.57 `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()`.

7.1.3.58 `template<class Model > void mappel::log_likelihood_stack (const Model & model, const typename Model::ImageT & image, const typename Model::ParamVecT & 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 128 of file stackcomp.h.

References `mappel::methods::objective::openmp::llh_stack()`, and `log_likelihood()`.

7.1.3.59 `template<class Model > void mappel::log_likelihood_stack (const Model & model, const typename Model::ImageStackT & image_stack, const typename Model::ParamVecT & 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 151 of file `stackcomp.h`.

References `mappel::methods::objective::openmp::llh_stack()`, and `log_likelihood()`.

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

Definition at line 248 of file `stencil.h`.

7.1.3.61 `double mappel::log_prior_beta_const (double beta) [inline]`

Definition at line 242 of file `stencil.h`.

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

Definition at line 255 of file `stencil.h`.

7.1.3.63 `double mappel::log_prior_normal_const (double sigma) [inline]`

Definition at line 267 of file `stencil.h`.

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

Definition at line 261 of file `stencil.h`.

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

Definition at line 97 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().

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

Definition at line 118 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().

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

Definition at line 133 of file stencil.h.

References fill_DXS2_stencil().

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

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

Definition at line 125 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().

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

Definition at line 141 of file stencil.h.

References fill_DXSX_stencil().

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

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

Definition at line 104 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().

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

Definition at line 216 of file stencil.h.

References fill_gaussian_stencil().

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

Definition at line 130 of file util.h.

References operator<<().

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

Definition at line 111 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().

7.1.3.74 int mappel::maxidx (const VecT & *v*)

Definition at line 55 of file util.cpp.

7.1.3.75 template<class Model > void mappel::model_grad_stack (const Model & *model*, const typename Model::ImageStackT & *image_stack*, const typename Model::ParamVecT & *theta_stack*, typename Model::ParamVecT & *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 188 of file stackcomp.h.

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

7.1.3.76 `template<class Model > void mappel::model_hessian_stack (const Model & model, const typename Model::ImageStackT & image_stack, const typename Model::ParamVecT & 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 231 of file stackcomp.h.

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

7.1.3.77 `template<class Model > void mappel::model_image_stack (const Model & model, const typename Model::ParamVecT & theta_stack, typename Model::ImageStackT & 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 62 of file stackcomp.h.

References `model_image()`.

7.1.3.78 `template<class Model > void mappel::model_positive_hessian_stack (const Model & model, const typename Model::ImageStackT & image_stack, const typename Model::ParamVecT & theta_stack, CubeT & hessian_stack)`

Parallel model positive-definite Hessian approximation calculations for a stack of images.

Compute Hessian a positive-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.
in	<i>theta_stack</i>	Sequence of thetas. Size: [model.num_params, nThetas]
out	<i>hess_stack</i>	Sequence of approximate Hessian matrices computed. Size: [model.num_params, model.num_params, n]

Definition at line 276 of file stackcomp.h.

7.1.3.79 `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 127 of file numerical.cpp.

Referenced by `cholesky_make_negative_definite()`, `cholesky_make_positive_definite()`, and `mappel::NewtonMaximizer< Model >::maximize()`.

7.1.3.80 `double mappel::normal_prior_grad (double mu, double sigma)`

7.1.3.81 `double mappel::normal_prior_grad (double sigma)`

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

Definition at line 332 of file stencil.h.

7.1.3.83 `double mappel::normal_prior_grad2 (double sigma)` `[inline]`

Definition at line 368 of file stencil.h.

7.1.3.84 `double mappel::normal_quantile_onesided (double confidence)`

Definition at line 29 of file stencil.cpp.

7.1.3.85 `double mappel::normal_quantile_twosided (double confidence)`

Definition at line 18 of file stencil.cpp.

Referenced by `mappel::methods::error_bounds_expected()`, `mappel::methods::openmp::error_bounds_expected_stack()`, and `mappel::methods::error_bounds_observed()`.

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

Definition at line 68 of file util.cpp.

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

Definition at line 276 of file PointEmitterModel.h.

7.1.3.88 `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::Gauss1DModel::Stencil::X`.

Referenced by `make_unique()`.

7.1.3.89 `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`, `TERM_CYAN`, `TERM_WHITE`, `mappel::Gauss1DsModel::Stencil::theta`, and `mappel::Gauss1DsModel::Stencil::X`.

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

Definition at line 226 of file estimator_impl.h.

7.1.3.91 `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`.

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

Definition at line 318 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`.

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

Definition at line 326 of file stencil.h.

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

Definition at line 362 of file stencil.h.

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

Definition at line 224 of file stencil.h.

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

7.1.3.96 `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()`.

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

7.1.3.98 `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()`.

Referenced by `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::estimate_max()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, and `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`.

7.1.3.99 `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()`.

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

Definition at line 154 of file display.cpp.

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

7.1.3.101 `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()`.

7.1.3.102 `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<<()`.

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

Definition at line 157 of file stencil.cpp.

References `gaussian_convolution()`.

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

Definition at line 225 of file stencil.cpp.

References gaussian_3D_convolution().

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

Definition at line 233 of file stencil.h.

Referenced by relative_log_likelihood(), mappel::methods::likelihood::rllh(), and mappel::methods::likelihood::debug←
::rllh_components().

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

Definition at line 123 of file util.h.

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

Definition at line 280 of file stencil.h.

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

Definition at line 273 of file stencil.h.

Referenced by mappel::PriorMAP1DObjective::prior_relative_log_likelihood().

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

Definition at line 287 of file stencil.h.

Referenced by mappel::PriorMAP1DObjective::prior_relative_log_likelihood().

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

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

Definition at line 299 of file stencil.h.

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

Definition at line 293 of file stencil.h.

7.1.3.113 `template<class Model> void mappel::sample_prior_stack (Model & model, typename Model::ParamVecT & 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 36 of file stackcomp.h.

References `rng_manager`.

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

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

Definition at line 116 of file util.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::compute_bound_scaling_vec()`.

7.1.3.115 `template<class Model > void mappel::simulate_image_stack (const Model & model, const typename Model::ParamVecT & theta_stack, typename Model::ImageStackT & 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 87 of file stackcomp.h.

References `model_image()`, `rng_manager`, `mappel::methods::simulate_image()`, and `mappel::methods::simulate_image_from_model()`.

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

Definition at line 121 of file util.h.

Referenced by `fill_DXS_stencil()`, and `fill_G_stencil()`.

7.1.4 Variable Documentation

7.1.4.1 `ParallelRngManagerT mappel::rng_manager`

Definition at line 6 of file rng.cpp.

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

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

Definition at line 13 of file display.cpp.

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

Definition at line 17 of file display.cpp.

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

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

Definition at line 19 of file display.cpp.

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

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

Definition at line 21 of file display.cpp.

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

Definition at line 25 of file display.cpp.

Referenced by `lambda_term_color()`.

7.1.4.7 `const char * mappel::TERM_DIM_CYAN ="0;36"`

Definition at line 27 of file display.cpp.

Referenced by `lambda_term_color()`.

7.1.4.8 `const char * mappel::TERM_DIM_GREEN = "0;32"`

Definition at line 23 of file display.cpp.

Referenced by `lambda_term_color()`.

7.1.4.9 `const char * mappel::TERM_DIM_MAGENTA = "0;35"`

Definition at line 26 of file display.cpp.

Referenced by `lambda_term_color()`.

7.1.4.10 `const char * mappel::TERM_DIM_RED = "0;31"`

Definition at line 22 of file display.cpp.

Referenced by `lambda_term_color()`.

7.1.4.11 `const char * mappel::TERM_DIM_WHITE = "0;37"`

Definition at line 28 of file display.cpp.

Referenced by `lambda_term_color()`.

7.1.4.12 `const char * mappel::TERM_DIM_YELLOW = "0;33"`

Definition at line 24 of file display.cpp.

Referenced by `lambda_term_color()`.

7.1.4.13 `const char * mappel::TERM_GREEN = "1;32"`

Definition at line 15 of file display.cpp.

Referenced by `lambda_term_color()`.

7.1.4.14 `const char * mappel::TERM_MAGENTA = "1;35"`

Definition at line 18 of file display.cpp.

Referenced by `lambda_term_color()`.

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

Definition at line 14 of file display.cpp.

Referenced by `lambda_term_color()`.

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

Definition at line 20 of file `display.cpp`.

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

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

Definition at line 16 of file `display.cpp`.

Referenced by `lambda_term_color()`.

7.2 mappel::mcmc Namespace Reference

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

7.2.1 Function Documentation

7.2.1.1 `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_mcmc_posterior()`.

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

Definition at line 7 of file `mcmc.cpp`.

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

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

Definition at line 32 of file `mcmc.h`.

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

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

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

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

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

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

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

Definition at line 13 of file `mcmc.cpp`.

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

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

Definition at line 24 of file `mcmc.cpp`.

7.3 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` (Model &model, const ParamT< Model > &theta)
- `template<class Model, class rng_t >`
`ModelDataT< Model > simulate_image` (Model &model, const ParamT< Model > &theta, rng_t &rng)
- `template<class Model >`
`ModelDataT< Model > simulate_image` (Model &model, const StencilT< Model > &s)
- `template<class Model >`
`ModelDataT< Model > simulate_image_from_model` (Model &model, const ImageT< Model > &model_im)
- `template<class Model >`
`void aposteriori_objective` (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`void aposteriori_objective` (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`void prior_objective` (const Model &model, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`void likelihood_objective` (const Model &model, const ModelDataT< Model > &data_im, const StencilT< Model > &s, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`void likelihood_objective` (const Model &model, const ModelDataT< Model > &data_im, const ParamT< Model > &theta, double &rllh, ParamT< Model > &grad, MatT &hess)
- `template<class Model >`
`ParamT< Model > cr_lower_bound` (const Model &model, const typename Model::Stencil &s)
Calculate the Cramer-Rao lower bound at the given parameters.
- `template<class Model >`
`ParamT< Model > cr_lower_bound` (const Model &model, const ParamT< Model > &theta)
- `template<class Model >`
`MatT expected_information` (const Model &model, const ParamT< Model > &theta)
- `template<class Model >`
`MatT observed_information` (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_mode)
- `template<class Model >`
`MatT observed_information` (const Model &model, const ModelDataT< Model > &data, const StencilT< Model > &theta_mode)
- `template<class Model >`
`StencilT< Model > estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method)
- `template<class Model >`
`StencilT< Model > estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, double &rllh)

- `template<class Model >`
`void estimate_max (Model &model, const ModelDataT< Model > &data, const std::string &method, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsI)`
- `template<class Model >`
`void estimate_max (Model &model, const ModelDataT< Model > &data, const std::string &method, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsI, StatsT &stats)`
- `template<class Model >`
`void estimate_max (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsI)`
- `template<class Model >`
`void estimate_max (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsI, StatsT &stats)`
- `template<class Model >`
`MatT estimate_mcmc_sample (Model &model, const ModelDataT< Model > &data, IdxT Nsample=1000, IdxT Nburnin=100, IdxT thin=0)`
- `template<class Model >`
`MatT estimate_mcmc_sample (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample=1000, IdxT Nburnin=100, IdxT thin=0)`
- `template<class Model >`
`void estimate_mcmc_sample (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample, IdxT Nburnin, IdxT thin, MatT &sample, VecT &sample_rllh)`
- `template<class Model >`
`void estimate_mcmc_posterior (Model &model, const ModelDataT< Model > &data, IdxT Nsample, IdxT Nburnin, IdxT thin, ParamT< Model > &posterior_mean, MatT &posterior_cov)`
- `template<class Model >`
`void estimate_mcmc_posterior (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample, IdxT Nburnin, IdxT thin, ParamT< Model > &posterior_mean, MatT &posterior_cov)`
- `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 ParamT< Model > &theta_est, MatT &obsI, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)`
- `template<class Model >`
`void error_bounds_posterior_credible (const Model &model, const MatT &sample, double confidence, ParamT< Model > &theta_mean, 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 (Model &model, const ParamT< Model > &theta)`
- `template<class Model, class RngT >`
`ModelDataT< Model > simulate_image (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< 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< Model > >, Model, PoissonNoise2DObjective > make_estimator` (Model &model, std::string ename)

7.3.1 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.

7.3.2 Function Documentation

7.3.2.1 `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 219 of file `model_methods_impl.h`.

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

Referenced by `aposteriori_objective()`.

7.3.2.2 `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)`

Definition at line 257 of file `model_methods_impl.h`.

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

7.3.2.3 `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 284 of file `model_methods_impl.h`.

References `expected_information()`.

Referenced by `cr_lower_bound()`, `mappel::cr_lower_bound_stack()`, and `error_bounds_expected()`.

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

Definition at line 296 of file `model_methods_impl.h`.

References `cr_lower_bound()`.

7.3.2.5 `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 440 of file `model_methods_impl.h`.

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

7.3.2.6 `template<class Model > void mappel::methods::error_bounds_observed (const Model & model, const ParamT< Model > & theta_est, MatT & obsl, double confidence, ParamT< Model > & theta_lb, ParamT< Model > & theta_ub)`

Definition at line 451 of file `model_methods_impl.h`.

References `mappel::normal_quantile_twosided()`.

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

Definition at line 467 of file `model_methods_impl.h`.

7.3.2.8 `template<class Model > StencilT< Model > mappel::methods::estimate_max (Model & model, const ModelDataT< Model > & data, const std::string & method)`

Definition at line 323 of file `model_methods_impl.h`.

References `make_estimator()`.

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

7.3.2.9 `template<class Model > StencilT< Model > mappel::methods::estimate_max (Model & model, const ModelDataT< Model > & data, const std::string & method, const ParamT< Model > & theta_init, double & rlh)`

Definition at line 330 of file `model_methods_impl.h`.

References `make_estimator()`.

7.3.2.10 `template<class Model > void mappel::methods::estimate_max (Model & model, const ModelDataT< Model > & data, const std::string & method, ParamT< Model > & theta_max, double & theta_max_llh, MatT & obsl)`

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

References `make_estimator()`.

7.3.2.11 `template<class Model > void mappel::methods::estimate_max (Model & model, const ModelDataT< Model > & data, const std::string & method, ParamT< Model > & theta_max, double & theta_max_llh, MatT & obsl, StatsT & stats)`

Definition at line 346 of file `model_methods_impl.h`.

References `make_estimator()`.

7.3.2.12 `template<class Model > void mappel::methods::estimate_max (Model & model, const ModelDataT< Model > & data, const std::string & method, const ParamT< Model > & theta_init, ParamT< Model > & theta_max, double & theta_max_llh, MatT & obsl)`

Definition at line 355 of file `model_methods_impl.h`.

References `make_estimator()`.

7.3.2.13 `template<class Model > void mappel::methods::estimate_max (Model & model, const ModelDataT< Model > & data, const std::string & method, const ParamT< Model > & theta_init, ParamT< Model > & theta_max, double & theta_max_llh, MatT & obsl, StatsT & stats)`

Definition at line 363 of file `model_methods_impl.h`.

References `make_estimator()`.

7.3.2.14 `template<class Model > void mappel::methods::estimate_mcmc_posterior (Model & model, const ModelDataT< Model > & data, IdxT Nsample, IdxT Nburnin, IdxT thin, ParamT< Model > & posterior_mean, MatT & posterior_cov)`

Definition at line 423 of file `model_methods_impl.h`.

Referenced by `estimate_mcmc_posterior()`.

7.3.2.15 `template<class Model > void mappel::methods::estimate_mcmc_posterior (Model & model, const ModelDataT< Model > & data, const ParamT< Model > & theta_init, IdxT Nsample, IdxT Nburnin, IdxT thin, ParamT< Model > & posterior_mean, MatT & posterior_cov)`

Definition at line 431 of file `model_methods_impl.h`.

References `estimate_mcmc_posterior()`, and `mappel::mcmc::estimate_sample_posterior()`.

7.3.2.16 `template<class Model > MatT mappel::methods::estimate_mcmc_sample (Model & model, const ModelDataT< Model > & data, IdxT Nsample = 1000, IdxT Nburnin = 100, IdxT thin = 0)`

Definition at line 386 of file `model_methods_impl.h`.

7.3.2.17 `template<class Model > MatT mappel::methods::estimate_mcmc_sample (Model & model, const ModelDataT< Model > & data, const ParamT< Model > & theta_init, IdxT Nsample = 1000, IdxT Nburnin = 100, IdxT thin = 0)`

Definition at line 393 of file `model_methods_impl.h`.

References `mappel::mcmc::num_oversample()`, `mappel::mcmc::sample_posterior()`, and `mappel::mcmc::thin_sample()`.

7.3.2.18 `template<class Model > void mappel::methods::estimate_mcmc_sample (Model & model, const ModelDataT< Model > & data, const ParamT< Model > & theta_init, IdxT Nsample, IdxT Nburnin, IdxT thin, MatT & sample, VecT & sample_rllh)`

Definition at line 406 of file `model_methods_impl.h`.

References `mappel::mcmc::num_oversample()`, `mappel::mcmc::sample_posterior()`, and `mappel::mcmc::thin_sample()`.

7.3.2.19 `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 θ

Returns

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

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

7.3.2.20 `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 θ). 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.

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

Definition at line 302 of file model_methods_impl.h.

Referenced by cr_lower_bound().

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

Definition at line 248 of file model_methods_impl.h.

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

Referenced by likelihood_objective().

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

Definition at line 271 of file model_methods_impl.h.

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

7.3.2.24 `template<class Model > ReturnIfSubclassT<std::unique_ptr<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(), and mappel::methods::openmp::estimate_profile_likelihood().

7.3.2.25 `template<class Model > ReturnIfSubclassT<std::unique_ptr<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()`.

7.3.2.26 `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 17 of file `model_methods_impl.h`.

References `model_image()`.

7.3.2.27 `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 17 of file `model_methods_impl.h`.

References `model_image()`.

7.3.2.28 `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`.

7.3.2.29 `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()`.

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

Definition at line 316 of file `model_methods_impl.h`.

Referenced by `mappel::Estimator< Model >::compute_estimate()`, and `mappel::Estimator< Model >::estimate_max←_debug()`.

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

Definition at line 308 of file `model_methods_impl.h`.

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

7.3.2.32 `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 230 of file `model_methods_impl.h`.

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

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

References `simulate_image()`.

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

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

Definition at line 30 of file `model_methods_impl.h`.

References `simulate_image()`.

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

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

References `simulate_image()`.

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

7.3.2.36 `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

<i>in</i>	<i>model</i>	Model object
<i>in</i>	<i>s</i>	The stencil computed at theta.
<i>in, out</i>	<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().

```
7.3.2.37 template<class Model , class rng_t > ReturnIfSubclassT<ImageT<Model>, Model, PoissonNoise2D↔
Objective> 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.

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

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

Definition at line 36 of file model_methods_impl.h.

References simulate_image().

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

Definition at line 42 of file model_methods_impl.h.

Referenced by mappel::simulate_image_stack().

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

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

Parameters

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

Returns

A simulated image corresponding to *model_im* under the noise model.

Definition at line 61 of file PoissonNoise1DObjective.h.

References `mappel::generate_poisson()`.

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

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

Parameters

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

Returns

A simulated image corresponding to *model_im* under the noise model.

Definition at line 64 of file PoissonNoise2DObjective.h.

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

7.4 mappel::methods::debug Namespace Reference

Functions

- `template<class Model >`
void `estimate_max_debug` (Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsI, [MatT](#) &sequence, [VecT](#) &sequence_rllh, [StatsT](#) &stats)
- `template<class Model >`
void `estimate_max_debug` (Model &model, const [ModelDataT](#)< Model > &data, const std::string &method, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsI, [MatT](#) &sequence, [VecT](#) &sequence_rllh, [StatsT](#) &stats)
- `template<class Model >`
void `estimate_mcmc_sample_debug` (Model &model, const [ModelDataT](#)< Model > &data, [IdxT](#) Nsample, [MatT](#) &sample, [VecT](#) &sample_rllh, [MatT](#) &candidates, [VecT](#) &candidates_rllh)
- `template<class Model >`
void `estimate_mcmc_sample_debug` (Model &model, const [ModelDataT](#)< Model > &data, const [ParamT](#)< Model > &theta_init, [IdxT](#) Nsample, [MatT](#) &sample, [VecT](#) &sample_rllh, [MatT](#) &candidates, [VecT](#) &candidates_rllh)

7.4.1 Function Documentation

7.4.1.1 `template<class Model > void mappel::methods::debug::estimate_max_debug (Model & model, const ModelDataT< Model > & data, const std::string & method, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh, StatsT & stats)`

Definition at line 483 of file `model_methods_impl.h`.

7.4.1.2 `template<class Model > void mappel::methods::debug::estimate_max_debug (Model & model, const ModelDataT< Model > & data, const std::string & method, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh, StatsT & stats)`

Definition at line 491 of file `model_methods_impl.h`.

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

7.4.1.3 `template<class Model > void mappel::methods::debug::estimate_mcmc_sample_debug (Model & model, const ModelDataT< Model > & data, IdxT Nsample, MatT & sample, VecT & sample_rllh, MatT & candidates, VecT & candidates_rllh)`

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

7.4.1.4 `template<class Model > void mappel::methods::debug::estimate_mcmc_sample_debug (Model & model, const ModelDataT< Model > & data, const ParamT< Model > & theta_init, IdxT Nsample, MatT & sample, VecT & sample_rllh, MatT & candidates, VecT & candidates_rllh)`

Definition at line 513 of file `model_methods_impl.h`.

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

7.5 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)

7.5.1 Function Documentation

7.5.1.1 `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`.

7.5.1.2 `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`.


```
7.5.1.3  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.

```
7.5.1.4  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.

```
7.5.1.5  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().

```
7.5.1.6  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.

```
7.5.1.7  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().

```
7.5.1.8  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.

7.5.1.9 `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()`.

7.5.1.10 `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`.

7.6 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)`

7.6.1 Function Documentation

7.6.1.1 `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.

7.6.1.2 `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`.

7.6.1.3 `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.

7.6.1.4 `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`.

7.6.1.5 `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()`.

7.6.1.6 `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`.

7.6.1.7 `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()`.

7.6.1.8 `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`.

7.7 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)`

7.7.1 Function Documentation

- 7.7.1.1 `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.

```
7.7.1.2 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::QuasiNewtonMaximizer< Model >::maximize(), mappel::model_grad_stack(), mappel::model_hessian_stack(), mappel::TrustRegionMaximizer< Model >::name(), negative_definite_hessian(), mappel::PriorMAP1DObjective::prior_grad_update(), and mappel::PriorMAP1DObjective::set_hyperparameters().

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

Definition at line 66 of file model_methods_impl.h.

References grad().

```
7.7.1.4 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.

```
7.7.1.5 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(), mappel::NewtonDiagonalMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >::name(), mappel::PriorMAP1DObjective::prior_grad2_update(), and mappel::PriorMAP1DObjective::set_hyperparameters().

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

Definition at line 78 of file model_methods_impl.h.

References grad2().

```
7.7.1.7 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 93 of file model_methods_impl.h.

References grad2().

7.7.1.8 `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.

7.7.1.9 `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::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, `negative_definite_hessian()`, and `mappel::methods::observed_information()`.

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

Definition at line 101 of file `model_methods_impl.h`.

References `hessian()`.

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

Definition at line 113 of file `model_methods_impl.h`.

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

7.7.1.12 `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 123 of file `model_methods_impl.h`.

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

7.7.1.13 `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 135 of file `model_methods_impl.h`.

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

7.7.1.14 `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()`.

7.7.1.15 `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.

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

Definition at line 50 of file model_methods_impl.h.

References llh().

7.7.1.17 `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 143 of file model_methods_impl.h.

Referenced by negative_definite_hessian(), and mappel::methods::objective::openmp::negative_definite_hessian_↵ stack().

7.7.1.18 `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 155 of file model_methods_impl.h.

References grad(), and negative_definite_hessian().

7.7.1.19 `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 165 of file model_methods_impl.h.

References grad(), and negative_definite_hessian().

7.7.1.20 `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 178 of file model_methods_impl.h.

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

7.7.1.21 `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.

7.7.1.22 `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::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::methods::aposteriori_objective()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::Estimator< Model >::estimate_max()`, `mappel::Estimator< Model >::estimate_max_debug()`, `mappel::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::methods::likelihood_objective()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, `mappel::HeuristicEstimator< Model >::name()`, `mappel::CGaussHeuristicEstimator< Model >::name()`, `mappel::CGaussMLE< Model >::name()`, `mappel::SimulatedAnnealingMaximizer< Model >::name()`, `mappel::PriorMAP1DObjective::prior_log_likelihood()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `relative_log_likelihood()`, `rllh()`, `mappel::methods::objective::openmp::rllh_stack()`, `mappel::mcmc::sample_posterior()`, `mappel::mcmc::sample_posterior_debug()`, and `mappel::Estimator< Model >::~Estimator()`.

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

Definition at line 58 of file model_methods_impl.h.

References `rllh()`.

7.8 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)

7.8.1 Function Documentation

7.8.1.1 `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.

7.8.1.2 `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()`.

7.8.1.3 `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 202 of file model_methods_impl.h.

References `grad_components()`.

7.8.1.4 `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.

7.8.1.5 `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()`.

7.8.1.6 **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 209 of file model_methods_impl.h.

References hessian_components().

7.8.1.7 **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.

7.8.1.8 **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().

7.8.1.9 **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 188 of file model_methods_impl.h.

References llh_components().

7.8.1.10 **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.

7.8.1.11 **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().

7.8.1.12 **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 195 of file model_methods_impl.h.

References rllh_components().

7.9 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.

7.9.1 Function Documentation

7.9.1.1 `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 542 of file `openmp_methods.h`.

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

7.9.1.2 `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 590 of file `openmp_methods.h`.

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

7.9.1.3 `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 406 of file openmp_methods.h.

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

Referenced by `mappel::log_likelihood_stack()`.

7.9.1.4 `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 431 of file openmp_methods.h.

References `mappel::methods::objective::llh()`, `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::run()`.

7.9.1.5 `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.
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 639 of file `openmp_methods.h`.

References `mappel::methods::objective::negative_definite_hessian()`, `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::run()`.

7.9.1.6 `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 478 of file `openmp_methods.h`.

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

Referenced by `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, and `mappel::Estimator< Model >::~~Estimator()`.

7.9.1.7 `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 514 of file `openmp_methods.h`.

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

Referenced by `rllh_stack()`.

7.10 mappel::methods::openmp Namespace Reference

Functions

- `template<class Model > void sample_prior_stack (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 (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 (Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_max_stack, VecT &theta_max_rllh, CubeT &obsI_stack)`
- `template<class Model > void estimate_max_stack (Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_max_stack, VecT &theta_max_rllh, CubeT &obsI_stack, StatsT &stats)`
- `template<class Model > void estimate_max_stack (Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_init_stack, ParamVecT< Model > &theta_max_stack, VecT &theta_max_rllh, CubeT &obsI_stack)`
- `template<class Model > void estimate_max_stack (Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_init_stack, ParamVecT< Model > &theta_max_stack, VecT &theta_max_rllh, CubeT &obsI_stack, StatsT &stats)`

- `template<class Model >`
`void estimate_profile_likelihood (Model &model, const ModelDataT< Model > &data, const IdxVecT &fixed_↔
parameters, const MatT &fixed_values, const std::string &method, VecT &profile_likelihood, ParamVecT< Model
> &profile_parameters)`
- `template<class Model >`
`void estimate_profile_likelihood (Model &model, const ModelDataT< Model > &data, const IdxVecT &fixed_↔
parameters, const MatT &fixed_values, const std::string &method, VecT &profile_likelihood, ParamVecT< Model
> &profile_parameters, StatsT &stats)`
- `template<class Model >`
`void estimate_profile_likelihood (Model &model, const ModelDataT< Model > &data, const IdxVecT &fixed_↔
parameters, const MatT &fixed_values, const std::string &method, const ParamVecT< Model > &theta_init, VecT
&profile_likelihood, ParamVecT< Model > &profile_parameters)`
- `template<class Model >`
`void estimate_profile_likelihood (Model &model, const ModelDataT< Model > &data, const IdxVecT &fixed_↔
parameters, const MatT &fixed_values, const std::string &method, const ParamVecT< Model > &theta_init, VecT
&profile_likelihood, ParamVecT< Model > &profile_parameters, StatsT &stats)`
- `template<class Model >`
`void estimate_mcmc_sample_stack (Model &model, const ModelDataStackT< Model > &data_stack, const
ParamVecT< Model > &theta_init_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, CubeT &sample_stack, MatT
&sample_rllh_stack)`
- `template<class Model >`
`void estimate_mcmc_sample_stack (Model &model, const ModelDataStackT< Model > &data_stack, IdxT
Nsamples, IdxT Nburnin, IdxT thin, CubeT &sample, MatT &sample_rllh)`
- `template<class Model >`
`void estimate_mcmc_posterior_stack (Model &model, const ModelDataStackT< Model > &data_stack, const
ParamVecT< Model > &theta_init_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, MatT &theta_mean_stack,
CubeT &theta_cov_stack)`
- `template<class Model >`
`void estimate_mcmc_posterior_stack (Model &model, const ModelDataStackT< Model > &data_stack, IdxT
Nsamples, IdxT Nburnin, IdxT thin, MatT &theta_mean_stack, CubeT &theta_cov_stack)`
- `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_posterior_credible_stack (const Model &model, const CubeT &sample_stack, double confi-
dence, MatT &theta_mean_stack, MatT &theta_lb_stack, MatT &theta_ub_stack)`

7.10.1 Function Documentation

7.10.1.1 `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 135 of file `openmp_methods.h`.

References `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::rethrow()`, and `omp_exception_↔
catcher::impl::OMPEXceptionCatcher< IntType >::run()`.

Referenced by `error_bounds_expected_stack()`.

7.10.1.2 `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 312 of file `openmp_methods.h`.

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

7.10.1.3 `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 327 of file `openmp_methods.h`.

References `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::run()`.

7.10.1.4 `template<class Model > void mappel::methods::openmp::error_bounds_posterior_credible_stack (const Model & model, const CubeT & sample_stack, double confidence, MatT & theta_mean_stack, MatT & theta_lb_stack, MatT & theta_ub_stack)`

Definition at line 358 of file `openmp_methods.h`.

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

7.10.1.5 `template<class Model > void mappel::methods::openmp::estimate_max_stack (Model & model, const ModelDataStackT< Model > & data_stack, const std::string & method, ParamVecT< Model > & theta_max_stack, VecT & theta_max_rllh, CubeT & obsl_stack)`

Definition at line 165 of file `openmp_methods.h`.

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

7.10.1.6 `template<class Model > void mappel::methods::openmp::estimate_max_stack (Model & model, const ModelDataStackT< Model > & data_stack, const std::string & method, ParamVecT< Model > & theta_max_stack, VecT & theta_max_rllh, CubeT & obsl_stack, StatsT & stats)`

Definition at line 173 of file `openmp_methods.h`.

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

7.10.1.7 `template<class Model > void mappel::methods::openmp::estimate_max_stack (Model & model, const ModelDataStackT< Model > & data_stack, const std::string & method, ParamVecT< Model > & theta_init_stack, ParamVecT< Model > & theta_max_stack, VecT & theta_max_rllh, CubeT & obsl_stack)`

Definition at line 182 of file `openmp_methods.h`.

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

7.10.1.8 `template<class Model > void mappel::methods::openmp::estimate_max_stack (Model & model, const ModelDataStackT< Model > & data_stack, const std::string & method, ParamVecT< Model > & theta_init_stack, ParamVecT< Model > & theta_max_stack, VecT & theta_max_rllh, CubeT & obsl_stack, StatsT & stats)`

Definition at line 190 of file `openmp_methods.h`.

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

7.10.1.9 `template<class Model > void mappel::methods::openmp::estimate_mcmc_posterior_stack (Model & model, const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, MatT & theta_mean_stack, CubeT & theta_cov_stack)`

Definition at line 275 of file `openmp_methods.h`.

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

Referenced by `estimate_mcmc_posterior_stack()`.

7.10.1.10 `template<class Model > void mappel::methods::openmp::estimate_mcmc_posterior_stack (Model & model, const ModelDataStackT< Model > & data_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, MatT & theta_mean_stack, CubeT & theta_cov_stack)`

Definition at line 303 of file `openmp_methods.h`.

References `estimate_mcmc_posterior_stack()`.

7.10.1.11 `template<class Model > void mappel::methods::openmp::estimate_mcmc_sample_stack (Model & model, const ModelDataStackT< Model > & data_stack, const ParamVecT< Model > & theta_init_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, CubeT & sample_stack, MatT & sample_rllh_stack)`

Definition at line 241 of file `openmp_methods.h`.

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

Referenced by `estimate_mcmc_sample_stack()`.

7.10.1.12 `template<class Model > void mappel::methods::openmp::estimate_mcmc_sample_stack (Model & model, const ModelDataStackT< Model > & data_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, CubeT & sample, MatT & sample_rllh)`

Definition at line 266 of file `openmp_methods.h`.

References `estimate_mcmc_sample_stack()`.

7.10.1.13 `template<class Model > void mappel::methods::openmp::estimate_profile_likelihood (Model & model, const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & fixed_values, const std::string & method, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters)`

Definition at line 200 of file `openmp_methods.h`.

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

7.10.1.14 `template<class Model > void mappel::methods::openmp::estimate_profile_likelihood (Model & model, const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & fixed_values, const std::string & method, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters, StatsT & stats)`

Definition at line 210 of file `openmp_methods.h`.

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

7.10.1.15 `template<class Model > void mappel::methods::openmp::estimate_profile_likelihood (Model & model, const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & fixed_values, const std::string & method, const ParamVecT< Model > & theta_init, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters)`

Definition at line 221 of file `openmp_methods.h`.

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

7.10.1.16 `template<class Model > void mappel::methods::openmp::estimate_profile_likelihood (Model & model, const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & fixed_values, const std::string & method, const ParamVecT< Model > & theta_init, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters, StatsT & stats)`

Definition at line 231 of file `openmp_methods.h`.

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

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

Definition at line 150 of file `openmp_methods.h`.

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

7.10.1.18 `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 70 of file openmp_methods.h.

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

7.10.1.19 `template<class Model > void mappel::methods::openmp::sample_prior_stack (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 42 of file openmp_methods.h.

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

7.10.1.20 `template<class Model > void mappel::methods::openmp::simulate_image_stack (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 97 of file openmp_methods.h.

References [mappel::methods::model_image\(\)](#), [omp_exception_catcher::impl_::OMPEExceptionCatcher< IntType >::rethrow\(\)](#), [omp_exception_catcher::impl_::OMPEExceptionCatcher< IntType >::run\(\)](#), and [mappel::methods::simulate_image\(\)](#).

7.11 omp_exception_catcher Namespace Reference

Namespaces

- [impl_](#)

Typedefs

- using [OMPEExceptionCatcher](#) = [impl_::OMPEExceptionCatcher< uint32_t >](#)

7.11.1 Typedef Documentation

7.11.1.1 using [omp_exception_catcher::OMPEExceptionCatcher](#) = typedef [impl_::OMPEExceptionCatcher<uint32_t>](#)

Definition at line 94 of file [OMPEExceptionCatcher.h](#).

7.12 omp_exception_catcher::impl_ Namespace Reference

Classes

- class [OMPEExceptionCatcher](#)

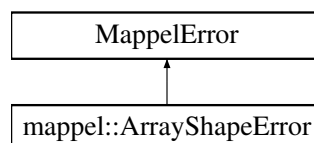
8 Class Documentation

8.1 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)

8.1.1 Detailed Description

Array is not of the right dimensionality.

Definition at line 72 of file util.h.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 mappel::ArrayShapeError::ArrayShapeError (std::string *message*) [inline]

Definition at line 74 of file util.h.

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

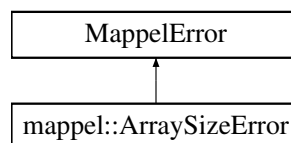
- [util.h](#)

8.2 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)

8.2.1 Detailed Description

Array is not of the right size.

Definition at line 79 of file util.h.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 mappel::ArraySizeError::ArraySizeError (std::string message) [inline]

Definition at line 81 of file util.h.

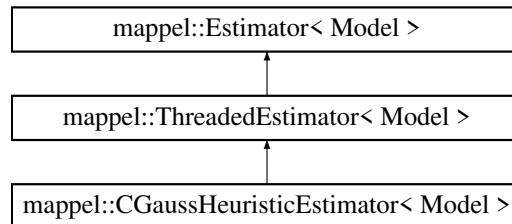
The documentation for this struct was generated from the following file:

- [util.h](#)

8.3 mappel::CGaussHeuristicEstimator< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::CGaussHeuristicEstimator< Model >:



Public Types

- enum [ExitCode](#) : IdxT {
[ExitCode::Unassigned](#) = 8, [ExitCode::MaxIter](#) = 7, [ExitCode::MaxBacktracks](#) = 6, [ExitCode::TrustRegionRadius](#) = 5,
[ExitCode::GradRatio](#) = 4, [ExitCode::FunctionChange](#) = 3, [ExitCode::StepSize](#) = 2, [ExitCode::Success](#) = 1,
[ExitCode::Error](#) = 0 }

Public Member Functions

- [CGaussHeuristicEstimator](#) (Model &model)
- [StatsT](#) [get_stats](#) ()
- [StatsT](#) [get_debug_stats](#) ()
- std::string [name](#) () const
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im, const [ParamVecT](#)< Model > &theta_init, [ParamVecT](#)< Model > &theta, [VecT](#) &rllh, [CubeT](#) &obsI)
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im_stack, [ParamVecT](#)< Model > &theta_est_← stack, [VecT](#) &rllh_stack, [CubeT](#) &obsI_stack)
- void [estimate_profile_stack](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed_parameters, const [MatT](#) &values, const [ParamVecT](#)< Model > &theta_init, [VecT](#) &profile_likelihood, [ParamVecT](#)< Model > &profile_← parameters)
- void [clear_stats](#) ()
- Model & [get_model](#) ()

- void `set_model` (Model &new_model)
- `StencilT`< Model > `estimate_max` (const `ModelDataT`< Model > &im)
- `StencilT`< Model > `estimate_max` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init)
- `StencilT`< Model > `estimate_max` (const `ModelDataT`< Model > &im, double &rllh)
- `StencilT`< Model > `estimate_max` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, double &rllh)
- void `estimate_max` (const `ModelDataT`< Model > &im, `ParamT`< Model > &theta, double &rllh, `MatT` &obsl)
- void `estimate_max` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, `ParamT`< Model > &theta, double &rllh, `MatT` &obsl)
- void `estimate_max_debug` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, `ParamT`< Model > &theta_est, double &rllh, `MatT` &obsl, `MatT` &sequence, `VecT` &sequence_rllh)

Static Public Attributes

- static constexpr int `NumExitCodes` = 9

Protected Member Functions

- void `record_exit_code` (`ExitCode` code)
- virtual void `compute_estimate` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, `ParamT`< Model > &theta_est, double &rllh, `MatT` &obsl)
Default base class implementation computes rllh and obsl separately from stencil This should be overridden by `Estimator` subclasses that already have access to this information.
- virtual `StencilT`< Model > `compute_estimate_debug` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, `ParamVecT`< Model > &sequence, `VecT` &sequence_rllh)
- virtual void `compute_profile_estimate` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, const `IdxVecT` &fixed_parameters, `ParamT`< Model > &theta_est, double &rllh)
- void `record_walltime` (`ClockT`::time_point start_walltime, int nimages)

Protected Attributes

- int `max_threads`
- int `num_threads`
- std::mutex `mtx`
- Model & `model`
- int `num_estimations` = 0
- double `total_walltime` = 0.
- `IdxVecT` `exit_counts`

8.3.1 Detailed Description

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

Definition at line 162 of file estimator.h.

8.3.2 Member Enumeration Documentation

8.3.2.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT` `[strong],[inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.3.3 Constructor & Destructor Documentation

8.3.3.1 `template<class Model > mappel::CGaussHeuristicEstimator< Model >::CGaussHeuristicEstimator (Model & model)` `[inline]`

Definition at line 164 of file estimator.h.

References `mappel::Estimator< Model >::get_debug_stats()`, and `mappel::Estimator< Model >::get_stats()`.

8.3.4 Member Function Documentation

8.3.4.1 `template<class Model > void mappel::ThreadedEstimator< Model >::clear_stats ()` `[virtual],[inherited]`

Reimplemented from [mappel::Estimator< Model >](#).

Reimplemented in [mappel::IterativeMaximizer< Model >](#).

Definition at line 378 of file estimator_impl.h.

References `mappel::Estimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::mtx`, and `mappel::ThreadedEstimator< Model >::num_threads`.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`.

```
8.3.4.2 template<class Model > void mappel::Estimator< Model >::compute_estimate ( const ModelDataT< Model
> & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl )
[protected], [virtual], [inherited]
```

Default base class implementation computes rllh and obsl separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::model](#), and [mappel::Estimator< Model >::methods::observed_information\(\)](#).

```
8.3.4.3 template<class Model > StencilT< Model > mappel::Estimator< Model >::compute_estimate_debug ( const
ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamVecT< Model > & sequence, VecT &
sequence_rllh ) [protected], [virtual], [inherited]
```

Estimators that produce a sequence of results (e.g. [IterativeEstimators](#)) can override this dummy debug implementation.

Reimplemented in [mappel::IterativeMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), and [mappel::CGaussMLE< Model >](#).

Definition at line 242 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::Error](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), and [mappel::Estimator< Model >::record_exit_code\(\)](#).

Referenced by [mappel::Estimator< Model >::estimate_max_debug\(\)](#), [mappel::CGaussMLE< Model >::name\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::name\(\)](#), and [mappel::Estimator< Model >::~~Estimator\(\)](#).

```
8.3.4.4 template<class Model > void mappel::Estimator< Model >::compute_profile_estimate ( const ModelDataT< Model
> & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT< Model > & theta_est,
double & rllh ) [protected], [virtual], [inherited]
```

Reimplemented in [mappel::IterativeMaximizer< Model >](#).

Definition at line 192 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), and [mappel::Estimator< Model >::~~Estimator\(\)](#).

```
8.3.4.5 template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max ( const ModelDataT<
Model > & im ) [inherited]
```

Definition at line 64 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.3.4.6 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init) [inherited]`

Definition at line 82 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::methods::objective::rllh()`.

8.3.4.7 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh) [inherited]`

Definition at line 73 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.3.4.8 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [inherited]`

Definition at line 90 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.3.4.9 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 119 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.3.4.10 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

8.3.4.11 `template<class Model > void mappel::Estimator< Model >::estimate_max_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

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

8.3.4.12 `template<class Model> void mappel::Estimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im_stack, ParamVecT< Model> & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack)`
`[inherited]`

Definition at line 170 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::estimate_max_stack()`, and `mappel::Estimator< Model>::model`.

8.3.4.13 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im, const ParamVecT< Model> & theta_init, ParamVecT< Model> & theta, VecT & rllh, CubeT & obsl)` `[virtual],[inherited]`

Implements `mappel::Estimator< Model>`.

Definition at line 285 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.3.4.14 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_profile_stack (const ModelDataT< Model> & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model> & theta_init, VecT & profile_likelihood, ParamVecT< Model> & profile_parameters)` `[virtual],[inherited]`

Implements `mappel::Estimator< Model>`.

Definition at line 323 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::record_walltime()`.

8.3.4.15 `template<class Model> StatsT mappel::CGaussHeuristicEstimator< Model>::get_debug_stats ()`
`[virtual]`

Reimplemented from `mappel::ThreadedEstimator< Model>`.

Definition at line 447 of file `estimator_impl.h`.

References `mappel::ThreadedEstimator< Model>::get_stats()`.

8.3.4.16 `template<class Model> Model & mappel::Estimator< Model>::get_model ()` `[inherited]`

Definition at line 50 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::model`.

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

8.3.4.17 `template<class Model> StatsT mappel::CGaussHeuristicEstimator< Model >::get_stats () [virtual]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 437 of file `estimator_impl.h`.

References [mappel::ThreadedEstimator< Model >::get_stats\(\)](#).

8.3.4.18 `template<class Model> std::string mappel::CGaussHeuristicEstimator< Model >::name () const [inline], [virtual]`

Implements [mappel::Estimator< Model >](#).

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

References [mappel::Estimator< Model >::compute_estimate\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

8.3.4.19 `template<class Model> void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code) [protected], [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 386 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::mtx](#), and [mappel::methods::objective::rllh\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::convergence_test\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

8.3.4.20 `template<class Model> void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages) [protected], [inherited]`

Definition at line 266 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::num_estimations](#), and [mappel::Estimator< Model >::total_walltime](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), [mappel::Estimator< Model >::estimate_max_debug\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), and [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#).

8.3.4.21 `template<class Model> void mappel::Estimator< Model >::set_model (Model & new_model) [inherited]`

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

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.3.5 Member Data Documentation

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

Definition at line 114 of file estimator.h.

Referenced by `mappel::Estimator< Model>::clear_stats()`, `mappel::Estimator< Model>::get_stats()`, and `mappel::ThreadedEstimator< Model>::record_exit_code()`.

8.3.5.2 `template<class Model> int mappel::ThreadedEstimator< Model>::max_threads` [protected], [inherited]

Definition at line 142 of file estimator.h.

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

Definition at line 109 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model>::anneal()`, `mappel::IterativeMaximizer< Model>::backtrack()`, `mappel::Estimator< Model>::compute_estimate()`, `mappel::CGaussMLE< Model>::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate()`, `mappel::IterativeMaximizer< Model>::compute_estimate()`, `mappel::Estimator< Model>::compute_estimate_debug()`, `mappel::CGaussMLE< Model>::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::Estimator< Model>::compute_profile_estimate()`, `mappel::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::estimate_max()`, `mappel::Estimator< Model>::estimate_max_debug()`, `mappel::Estimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model>::get_debug_stats()`, `mappel::Estimator< Model>::get_model()`, `mappel::IterativeMaximizer< Model>::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model>::maximize()`, `mappel::NewtonMaximizer< Model>::maximize()`, `mappel::QuasiNewtonMaximizer< Model>::maximize()`, `mappel::TrustRegionMaximizer< Model>::maximize()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::set_model()`.

8.3.5.4 `template<class Model> std::mutex mappel::ThreadedEstimator< Model>::mtx` [protected], [inherited]

Definition at line 144 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model>::clear_stats()`, `mappel::IterativeMaximizer< Model>::clear_stats()`, `mappel::ThreadedEstimator< Model>::get_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::IterativeMaximizer< Model>::record_run_statistics()`.

8.3.5.5 `template<class Model> int mappel::Estimator< Model>::num_estimations = 0` [protected], [inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model>::clear_stats()`, `mappel::Estimator< Model>::get_stats()`, `mappel::ThreadedEstimator< Model>::get_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, and `mappel::Estimator< Model>::record_walltime()`.

8.3.5.6 `template<class Model > int mappel::ThreadedEstimator< Model >::num_threads` [protected],
[inherited]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.3.5.7 `template<class Model > constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static],
[inherited]

Definition at line 39 of file estimator.h.

8.3.5.8 `template<class Model > double mappel::Estimator< Model >::total_walltime = 0.` [protected],
[inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

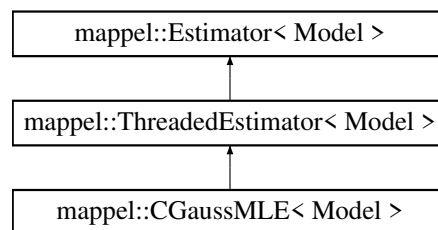
The documentation for this class was generated from the following files:

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

8.4 mappel::CGaussMLE< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::CGaussMLE< Model >`:



Public Types

- enum `ExitCode` : `IdxT` {
`ExitCode::Unassigned = 8`, `ExitCode::MaxIter = 7`, `ExitCode::MaxBacktracks = 6`, `ExitCode::TrustRegionRadius = 5`,
`ExitCode::GradRatio = 4`, `ExitCode::FunctionChange = 3`, `ExitCode::StepSize = 2`, `ExitCode::Success = 1`,
`ExitCode::Error = 0` }

Public Member Functions

- [CGaussMLE](#) (Model &model, int num_iterations=DEFAULT_CGAUSS_ITERATIONS)
- [StatsT get_stats](#) ()
- [StatsT get_debug_stats](#) ()
- [std::string name](#) () const
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im, const [ParamVecT](#)< Model > &theta_init, [ParamVecT](#)< Model > &theta, [VecT](#) &rllh, [CubeT](#) &obsI)
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im_stack, [ParamVecT](#)< Model > &theta_est_← stack, [VecT](#) &rllh_stack, [CubeT](#) &obsI_stack)
- void [estimate_profile_stack](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed_parameters, const [MatT](#) &values, const [ParamVecT](#)< Model > &theta_init, [VecT](#) &profile_likelihood, [ParamVecT](#)< Model > &profile_← parameters)
- void [clear_stats](#) ()
- Model & [get_model](#) ()
- void [set_model](#) (Model &new_model)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, double &rllh)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, double &rllh)
- void [estimate_max](#) (const [ModelDataT](#)< Model > &im, [ParamT](#)< Model > &theta, double &rllh, [MatT](#) &obsI)
- void [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta, double &rllh, [MatT](#) &obsI)
- void [estimate_max_debug](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsI, [MatT](#) &sequence, [VecT](#) &sequence_rllh)

Public Attributes

- int [num_iterations](#)

Static Public Attributes

- static constexpr int [NumExitCodes](#) = 9

Protected Member Functions

- [StencilT](#)< Model > [compute_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, double &rllh)
 - [StencilT](#)< Model > [compute_estimate_debug](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamVecT](#)< Model > &sequence, [VecT](#) &sequence_rllh)
 - void [record_exit_code](#) (ExitCode code)
 - virtual void [compute_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsI)
- Default base class implementation computes rllh and obsI separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.*
- virtual void [compute_profile_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, const [IdxVecT](#) &fixed_parameters, [ParamT](#)< Model > &theta_est, double &rllh)
 - void [record_walltime](#) (ClockT::time_point start_walltime, int nimages)

Protected Attributes

- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.
- [IdxVecT](#) [exit_counts](#)

8.4.1 Detailed Description

```
template<class Model>
class mappel::CGaussMLE< Model >
```

Definition at line 176 of file estimator.h.

8.4.2 Member Enumeration Documentation

8.4.2.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT` `[strong],[inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.4.3 Constructor & Destructor Documentation

8.4.3.1 `template<class Model > mappel::CGaussMLE< Model >::CGaussMLE (Model & model, int num_iterations =`
`DEFAULT_CGAUSS_ITERATIONS)` `[inline]`

Definition at line 179 of file estimator.h.

References `mappel::Estimator< Model >::get_debug_stats()`, and `mappel::Estimator< Model >::get_stats()`.

8.4.4 Member Function Documentation

8.4.4.1 `template<class Model > void mappel::ThreadedEstimator< Model >::clear_stats () [virtual], [inherited]`

Reimplemented from [mappel::Estimator< Model >](#).

Reimplemented in [mappel::IterativeMaximizer< Model >](#).

Definition at line 378 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::clear_stats\(\)](#), [mappel::ThreadedEstimator< Model >::mtx](#), and [mappel::ThreadedEstimator< Model >::num_threads](#).

Referenced by [mappel::IterativeMaximizer< Model >::clear_stats\(\)](#).

8.4.4.2 `template<class Model > void mappel::Estimator< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl) [protected], [virtual], [inherited]`

Default base class implementation computes rllh and obsl separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::model](#), and [mappel::methods::observed_information\(\)](#).

8.4.4.3 `template<class Model > StencilT< Model > mappel::CGaussMLE< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [protected], [virtual]`

Implements [mappel::Estimator< Model >](#).

Definition at line 471 of file `estimator_impl.h`.

References [mappel::cgauss_compute_estimate\(\)](#), [mappel::cgauss_heuristic_compute_estimate\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

8.4.4.4 `template<class Model > StencilT< Model > mappel::CGaussMLE< Model >::compute_estimate_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamVecT< Model > & sequence, VecT & sequence_rllh) [protected], [virtual]`

Estimators that produce a sequence of results (e.g. [IterativeEstimators](#)) can override this dummy debug implementation.

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 498 of file `estimator_impl.h`.

References [mappel::cgauss_compute_estimate_debug\(\)](#), [mappel::cgauss_heuristic_compute_estimate\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), and [mappel::methods::objective::openmp::rllh_stack\(\)](#).

8.4.4.5 `template<class Model > void mappel::Estimator< Model >::compute_profile_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT< Model > & theta_est, double & rllh)` [protected], [virtual], [inherited]

Reimplemented in [mappel::IterativeMaximizer< Model >](#).

Definition at line 192 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.4.4.6 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im)` [inherited]

Definition at line 64 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.4.4.7 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init)` [inherited]

Definition at line 82 of file estimator_impl.h.

References [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

8.4.4.8 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh)` [inherited]

Definition at line 73 of file estimator_impl.h.

References [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::model](#).

8.4.4.9 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh)` [inherited]

Definition at line 90 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::Error](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), [mappel::Estimator< Model >::record_exit_code\(\)](#), and [mappel::Estimator< Model >::record_walltime\(\)](#).

8.4.4.10 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl)` [inherited]

Definition at line 119 of file estimator_impl.h.

References [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::model](#).

8.4.4.11 `template<class Model> void mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::print_image()`, `mappel::Estimator< Model>::record_exit_code()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::rllh()`.

8.4.4.12 `template<class Model> void mappel::Estimator< Model>::estimate_max_debug (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate_debug()`, `mappel::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::rllh()`.

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

8.4.4.13 `template<class Model> void mappel::Estimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im_stack, ParamVecT< Model> & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack) [inherited]`

Definition at line 170 of file estimator_impl.h.

References `mappel::Estimator< Model>::estimate_max_stack()`, and `mappel::Estimator< Model>::model`.

8.4.4.14 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im, const ParamVecT< Model> & theta_init, ParamVecT< Model> & theta, VecT & rllh, CubeT & obsl) [virtual], [inherited]`

Implements [mappel::Estimator< Model>](#).

Definition at line 285 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.4.4.15 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_profile_stack (const ModelDataT< Model> & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model> & theta_init, VecT & profile_likelihood, ParamVecT< Model> & profile_parameters) [virtual], [inherited]`

Implements [mappel::Estimator< Model>](#).

Definition at line 323 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::record_walltime()`.

8.4.4.16 `template<class Model> StatsT mappel::CGaussMLE< Model >::get_debug_stats () [virtual]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 464 of file `estimator_impl.h`.

References [mappel::ThreadedEstimator< Model >::get_stats\(\)](#).

8.4.4.17 `template<class Model> Model & mappel::Estimator< Model >::get_model () [inherited]`

Definition at line 50 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.4.4.18 `template<class Model> StatsT mappel::CGaussMLE< Model >::get_stats () [virtual]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 454 of file `estimator_impl.h`.

References [mappel::ThreadedEstimator< Model >::get_stats\(\)](#).

8.4.4.19 `template<class Model> std::string mappel::CGaussMLE< Model >::name () const [inline],[virtual]`

Implements [mappel::Estimator< Model >](#).

Definition at line 184 of file `estimator.h`.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::compute_estimate_↵ debug\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

8.4.4.20 `template<class Model> void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code) [protected],[virtual],[inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 386 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::model](#), [mappel::Threaded↵ Estimator< Model >::mtx](#), and [mappel::methods::objective::rllh\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::CGaussMLE< Model >::compute_↵ _estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::convergence_test\(\)](#), [mappel::Threaded↵ Estimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

8.4.4.21 `template<class Model> void mappel::Estimator< Model>::record_walltime (ClockT::time_point start_walltime, int nimages)` `[protected]`, `[inherited]`

Definition at line 266 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::num_estimations`, and `mappel::Estimator< Model>::total_walltime`.

Referenced by `mappel::Estimator< Model>::estimate_max()`, `mappel::Estimator< Model>::estimate_max_↵ debug()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, and `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`.

8.4.4.22 `template<class Model> void mappel::Estimator< Model>::set_model (Model & new_model)` `[inherited]`

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

References `mappel::Estimator< Model>::model`.

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

8.4.5 Member Data Documentation

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

Definition at line 114 of file `estimator.h`.

Referenced by `mappel::Estimator< Model>::clear_stats()`, `mappel::Estimator< Model>::get_stats()`, and `mappel::↵ ThreadedEstimator< Model>::record_exit_code()`.

8.4.5.2 `template<class Model> int mappel::ThreadedEstimator< Model>::max_threads` `[protected]`, `[inherited]`

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

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

Definition at line 109 of file `estimator.h`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model>::anneal()`, `mappel::IterativeMaximizer< Model>::backtrack()`, `mappel::Estimator< Model>::compute_estimate()`, `mappel::CGaussMLE< Model>::compute_↵ _estimate()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate()`, `mappel::IterativeMaximizer< Model>::compute_estimate()`, `mappel::Estimator< Model>::compute_estimate_debug()`, `mappel::CGaussML↵ E< Model>::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate_↵ debug()`, `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::Estimator< Model>::compute_↵ _profile_estimate()`, `mappel::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::estimate_max()`, `mappel::Estimator< Model>::estimate_max_debug()`, `mappel::Estimator< Model>::estimate_↵ _max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model>::get_debug_stats()`, `mappel::Estimator< Model>::get_model()`, `mappel::IterativeMaximizer< Model>::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model>::maximize()`, `mappel::NewtonMaximizer< Model>::maximize()`, `mappel::QuasiNewtonMaximizer< Model>↵ ::maximize()`, `mappel::TrustRegionMaximizer< Model>::maximize()`, `mappel::ThreadedEstimator< Model>::record_↵ _exit_code()`, and `mappel::Estimator< Model>::set_model()`.

8.4.5.4 `template<class Model > std::mutex mappel::ThreadedEstimator< Model >::mtx` [protected],
[inherited]

Definition at line 144 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.4.5.5 `template<class Model > int mappel::Estimator< Model >::num_estimations = 0` [protected],
[inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

8.4.5.6 `template<class Model > int mappel::CGaussMLE< Model >::num_iterations`

Definition at line 178 of file estimator.h.

8.4.5.7 `template<class Model > int mappel::ThreadedEstimator< Model >::num_threads` [protected],
[inherited]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.4.5.8 `template<class Model > constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static],
[inherited]

Definition at line 39 of file estimator.h.

8.4.5.9 `template<class Model > double mappel::Estimator< Model >::total_walltime = 0.` [protected],
[inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

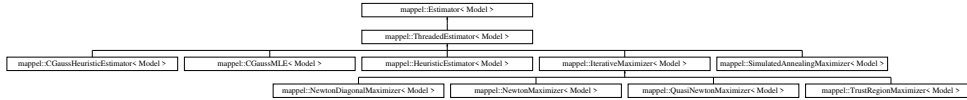
The documentation for this class was generated from the following files:

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

8.5 mappel::Estimator< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::Estimator< Model >:



Public Types

- enum `ExitCode` : `IdxT` {
`ExitCode::Unassigned` = 8, `ExitCode::MaxIter` = 7, `ExitCode::MaxBacktracks` = 6, `ExitCode::TrustRegionRadius` = 5,
`ExitCode::GradRatio` = 4, `ExitCode::FunctionChange` = 3, `ExitCode::StepSize` = 2, `ExitCode::Success` = 1,
`ExitCode::Error` = 0 }

Public Member Functions

- `Estimator` (`Model &_model`)
- virtual `~Estimator` ()
- virtual `std::string name` () const =0
- `Model & get_model` ()
- void `set_model` (`Model &new_model`)
- `StencilT< Model > estimate_max` (const `ModelDataT< Model > &im`)
- `StencilT< Model > estimate_max` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`)
- `StencilT< Model > estimate_max` (const `ModelDataT< Model > &im`, double `&rllh`)
- `StencilT< Model > estimate_max` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, double `&rllh`)
- void `estimate_max` (const `ModelDataT< Model > &im`, `ParamT< Model > &theta`, double `&rllh`, `MatT &obsI`)
- void `estimate_max` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, `ParamT< Model > &theta`, double `&rllh`, `MatT &obsI`)
- void `estimate_max_debug` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, `ParamT< Model > &theta_est`, double `&rllh`, `MatT &obsI`, `MatT &sequence`, `VecT &sequence_rllh`)
- virtual void `estimate_max_stack` (const `ModelDataStackT< Model > &im_stack`, const `ParamVecT< Model > &theta_init_stack`, `ParamVecT< Model > &theta_est_stack`, `VecT &rllh_stack`, `CubeT &obsI_stack`)=0
- void `estimate_max_stack` (const `ModelDataStackT< Model > &im_stack`, `ParamVecT< Model > &theta_est_stack`, `VecT &rllh_stack`, `CubeT &obsI_stack`)
- virtual void `estimate_profile_stack` (const `ModelDataT< Model > &data`, const `IdxVecT &fixed_parameters`, const `MatT &values`, const `ParamVecT< Model > &theta_init`, `VecT &profile_likelihood`, `ParamVecT< Model > &profile_parameters`)=0
- virtual `StatsT get_stats` ()
- virtual `StatsT get_debug_stats` ()=0
- virtual void `clear_stats` ()

Static Public Attributes

- static constexpr int `NumExitCodes` = 9

Protected Member Functions

- virtual [StencilT](#)< Model > [compute_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, double &rllh)=0
- virtual [StencilT](#)< Model > [compute_estimate_debug](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamVecT](#)< Model > &sequence, [VecT](#) &sequence_rllh)
- virtual void [compute_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsI)
Default base class implementation computes rllh and obsI separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.
- virtual void [compute_profile_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, const [IdxVecT](#) &fixed_parameters, [ParamT](#)< Model > &theta_est, double &rllh)
- void [record_walltime](#) (ClockT::time_point start_walltime, int nimages)
- virtual void [record_exit_code](#) ([ExitCode](#) code)=0

Protected Attributes

- 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)

8.5.1 Detailed Description

```
template<class Model>
class mappel::Estimator< Model >
```

Definition at line 37 of file estimator.h.

8.5.2 Member Enumeration Documentation

8.5.2.1 template<class Model > enum mappel::Estimator::ExitCode : IdxT [strong]

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.5.3 Constructor & Destructor Documentation

8.5.3.1 `template<class Model > mappel::Estimator< Model >::Estimator (Model & _model)`

Definition at line 44 of file estimator_impl.h.

8.5.3.2 `template<class Model > virtual mappel::Estimator< Model >::~Estimator () [inline], [virtual]`

Definition at line 52 of file estimator.h.

References `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::estimate_max()`, `mappel::Estimator< Model >::estimate_max_debug()`, `mappel::Estimator< Model >::estimate_max_stack()`, `mappel::Estimator< Model >::estimate_profile_stack()`, `mappel::Estimator< Model >::get_debug_stats()`, `mappel::Estimator< Model >::get_model()`, `mappel::Estimator< Model >::get_stats()`, `mappel::Estimator< Model >::name()`, `mappel::methods::objective::rllh()`, `mappel::methods::objective::openmp::rllh_stack()`, and `mappel::Estimator< Model >::set_model()`.

8.5.4 Member Function Documentation

8.5.4.1 `template<class Model > void mappel::Estimator< Model >::clear_stats () [virtual]`

Reimplemented in [mappel::IterativeMaximizer< Model >](#), and [mappel::ThreadedEstimator< Model >](#).

Definition at line 218 of file estimator_impl.h.

References `mappel::Estimator< Model >::exit_counts`, `mappel::Estimator< Model >::num_estimations`, and `mappel::Estimator< Model >::total_walltime`.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, and `mappel::Estimator< Model >::~Estimator()`.

8.5.4.2 `template<class Model > virtual StencilT<Model> mappel::Estimator< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [protected], [pure virtual]`

Implemented in [mappel::IterativeMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), and [mappel::CGaussMLE< Model >](#).

Referenced by `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::estimate_max()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::HeuristicEstimator< Model >::name()`, `mappel::CGaussHeuristicEstimator< Model >::name()`, `mappel::CGaussMLE< Model >::name()`, `mappel::SimulatedAnnealingMaximizer< Model >::name()`, and `mappel::Estimator< Model >::~Estimator()`.

```
8.5.4.3 template<class Model > void mappel::Estimator< Model >::compute_estimate ( const ModelDataT< Model
> & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl )
[protected], [virtual]
```

Default base class implementation computes rllh and obsl separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::model](#), and [mappel::Estimator< Model >::methods::observed_information\(\)](#).

```
8.5.4.4 template<class Model > StencilT< Model > mappel::Estimator< Model >::compute_estimate_debug ( const
ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamVecT< Model > & sequence, VecT &
sequence_rllh ) [protected], [virtual]
```

Estimators that produce a sequence of results (e.g. [IterativeEstimators](#)) can override this dummy debug implementation.

Reimplemented in [mappel::IterativeMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), and [mappel::CGaussMLE< Model >](#).

Definition at line 242 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::Error](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), and [mappel::Estimator< Model >::record_exit_code\(\)](#).

Referenced by [mappel::Estimator< Model >::estimate_max_debug\(\)](#), [mappel::CGaussMLE< Model >::name\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::name\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

```
8.5.4.5 template<class Model > void mappel::Estimator< Model >::compute_profile_estimate ( const ModelDataT< Model
> & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT< Model > & theta_est,
double & rllh ) [protected], [virtual]
```

Reimplemented in [mappel::IterativeMaximizer< Model >](#).

Definition at line 192 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

```
8.5.4.6 template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max ( const ModelDataT<
Model > & im )
```

Definition at line 64 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.5.4.7 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init)`

Definition at line 82 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::methods::objective::rllh()`.

8.5.4.8 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh)`

Definition at line 73 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.5.4.9 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh)`

Definition at line 90 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.5.4.10 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl)`

Definition at line 119 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.5.4.11 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta, double & rllh, MatT & obsl)`

Definition at line 128 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

8.5.4.12 `template<class Model > void mappel::Estimator< Model >::estimate_max_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh)`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

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

8.5.4.13 `template<class Model > virtual void mappel::Estimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im_stack, const ParamVecT< Model > & theta_init_stack, ParamVecT< Model > & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack) [pure virtual]`

Implemented in [mappel::ThreadedEstimator< Model >](#).

Referenced by [mappel::Estimator< Model >::estimate_max_stack\(\)](#), and [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.5.4.14 `template<class Model > void mappel::Estimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im_stack, ParamVecT< Model > & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack)`

Definition at line 170 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::estimate_max_stack\(\)](#), and [mappel::Estimator< Model >::model](#).

8.5.4.15 `template<class Model > virtual void mappel::Estimator< Model >::estimate_profile_stack (const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model > & theta_init, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters) [pure virtual]`

Implemented in [mappel::ThreadedEstimator< Model >](#).

Referenced by [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.5.4.16 `template<class Model > virtual StatsT mappel::Estimator< Model >::get_debug_stats () [pure virtual]`

Implemented in [mappel::IterativeMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), [mappel::CGaussMLE< Model >](#), [mappel::CGaussHeuristicEstimator< Model >](#), [mappel::HeuristicEstimator< Model >](#), and [mappel::ThreadedEstimator< Model >](#).

Referenced by [mappel::CGaussHeuristicEstimator< Model >::CGaussHeuristicEstimator\(\)](#), [mappel::CGaussMLE< Model >::CGaussMLE\(\)](#), [mappel::HeuristicEstimator< Model >::HeuristicEstimator\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::SimulatedAnnealingMaximizer\(\)](#), and [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.5.4.17 `template<class Model > Model & mappel::Estimator< Model >::get_model ()`

Definition at line 50 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.5.4.18 `template<class Model > StatsT mappel::Estimator< Model >::get_stats () [virtual]`

Reimplemented in [mappel::IterativeMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), [mappel::CGaussMLE< Model >](#), [mappel::CGaussHeuristicEstimator< Model >](#), [mappel::HeuristicEstimator< Model >](#), and [mappel::ThreadedEstimator< Model >](#).

Definition at line 200 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::Error](#), [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::FunctionChange](#), [mappel::Estimator< Model >::GradRatio](#), [mappel::Estimator< Model >::MaxBacktracks](#), [mappel::Estimator< Model >::MaxIter](#), [mappel::Estimator< Model >::num_estimations](#), [mappel::Estimator< Model >::StepSize](#), [mappel::Estimator< Model >::Success](#), [mappel::Estimator< Model >::total_walltime](#), and [mappel::Estimator< Model >::TrustRegionRadius](#).

Referenced by [mappel::CGaussHeuristicEstimator< Model >::CGaussHeuristicEstimator\(\)](#), [mappel::CGaussMLE< Model >::CGaussMLE\(\)](#), [mappel::ThreadedEstimator< Model >::get_stats\(\)](#), [mappel::HeuristicEstimator< Model >::HeuristicEstimator\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::SimulatedAnnealingMaximizer\(\)](#), and [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.5.4.19 `template<class Model> virtual std::string mappel::Estimator< Model>::name () const` `[pure virtual]`

Implemented in [mappel::TrustRegionMaximizer< Model>](#), [mappel::QuasiNewtonMaximizer< Model>](#), [mappel::NewtonMaximizer< Model>](#), [mappel::NewtonDiagonalMaximizer< Model>](#), [mappel::SimulatedAnnealingMaximizer< Model>](#), [mappel::CGaussMLE< Model>](#), [mappel::CGaussHeuristicEstimator< Model>](#), and [mappel::HeuristicEstimator< Model>](#).

Referenced by [mappel::Estimator< Model>::~~Estimator\(\)](#).

8.5.4.20 `template<class Model> virtual void mappel::Estimator< Model>::record_exit_code (ExitCode code)`
`[protected],[pure virtual]`

Implemented in [mappel::ThreadedEstimator< Model>](#).

Referenced by [mappel::Estimator< Model>::compute_estimate_debug\(\)](#), and [mappel::Estimator< Model>::estimate_max\(\)](#).

8.5.4.21 `template<class Model> void mappel::Estimator< Model>::record_walltime (ClockT::time_point start_walltime, int nimages)` `[protected]`

Definition at line 266 of file [estimator_impl.h](#).

References [mappel::Estimator< Model>::num_estimations](#), and [mappel::Estimator< Model>::total_walltime](#).

Referenced by [mappel::Estimator< Model>::estimate_max\(\)](#), [mappel::Estimator< Model>::estimate_max_debug\(\)](#), [mappel::ThreadedEstimator< Model>::estimate_max_stack\(\)](#), and [mappel::ThreadedEstimator< Model>::estimate_profile_stack\(\)](#).

8.5.4.22 `template<class Model> void mappel::Estimator< Model>::set_model (Model & new_model)`

Definition at line 54 of file [estimator_impl.h](#).

References [mappel::Estimator< Model>::model](#).

Referenced by [mappel::Estimator< Model>::~~Estimator\(\)](#).

8.5.5 Friends And Related Function Documentation

8.5.5.1 `template<class Model> template<class T> std::ostream& operator<< (std::ostream & out, Estimator< T> & estimator)` `[friend]`

8.5.6 Member Data Documentation

8.5.6.1 `template<class Model> IdxVecT mappel::Estimator< Model>::exit_counts` `[protected]`

Definition at line 114 of file [estimator.h](#).

Referenced by [mappel::Estimator< Model>::clear_stats\(\)](#), [mappel::Estimator< Model>::get_stats\(\)](#), and [mappel::ThreadedEstimator< Model>::record_exit_code\(\)](#).

8.5.6.2 `template<class Model> Model& mappel::Estimator< Model>::model` [protected]

Definition at line 109 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model>::anneal()`, `mappel::IterativeMaximizer< Model>::backtrack()`, `mappel::Estimator< Model>::compute_estimate()`, `mappel::CGaussMLE< Model>::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate()`, `mappel::IterativeMaximizer< Model>::compute_estimate()`, `mappel::Estimator< Model>::compute_estimate_debug()`, `mappel::CGaussMLE< Model>::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::Estimator< Model>::compute_profile_estimate()`, `mappel::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::estimate_max()`, `mappel::Estimator< Model>::estimate_max_debug()`, `mappel::Estimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model>::get_debug_stats()`, `mappel::Estimator< Model>::get_model()`, `mappel::IterativeMaximizer< Model>::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model>::maximize()`, `mappel::NewtonMaximizer< Model>::maximize()`, `mappel::QuasiNewtonMaximizer< Model>::maximize()`, `mappel::TrustRegionMaximizer< Model>::maximize()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::set_model()`.

8.5.6.3 `template<class Model> int mappel::Estimator< Model>::num_estimations = 0` [protected]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model>::clear_stats()`, `mappel::Estimator< Model>::get_stats()`, `mappel::ThreadedEstimator< Model>::get_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, and `mappel::Estimator< Model>::record_walltime()`.

8.5.6.4 `template<class Model> constexpr int mappel::Estimator< Model>::NumExitCodes = 9` [static]

Definition at line 39 of file estimator.h.

8.5.6.5 `template<class Model> double mappel::Estimator< Model>::total_walltime = 0.` [protected]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model>::clear_stats()`, `mappel::Estimator< Model>::get_stats()`, `mappel::ThreadedEstimator< Model>::get_stats()`, and `mappel::Estimator< Model>::record_walltime()`.

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

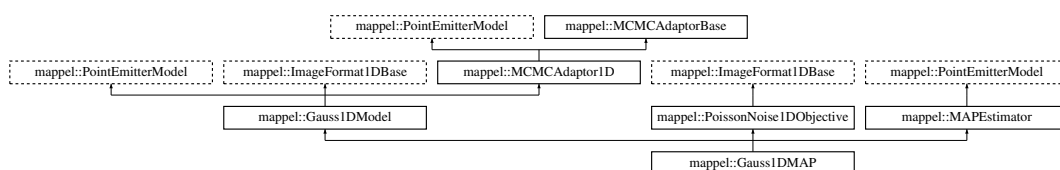
- [estimator.h](#)
- [estimator_impl.h](#)

8.6 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)`
- `ParamT sample_prior ()`
- `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)
- void `set_intensity_mcmc_sampling` (double eta_l=-1)
- void `set_background_mcmc_sampling` (double eta_bg=-1)
- void `set_mcmc_sigma_scale` (double scale)
- double `get_mcmc_sigma_scale` () const
- `IdxT get_mcmc_num_phases` () const

Static Public Member Functions

- static `CompositeDist` `make_default_prior` (`IdxT` size, const std::string &prior_type)
 - static `CompositeDist` `make_default_prior_beta_position` (`IdxT` size)
 - static `CompositeDist` `make_default_prior_normal_position` (`IdxT` size)
 - static `CompositeDist` `make_prior_beta_position` (`IdxT` size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
 - static `CompositeDist` `make_prior_normal_position` (`IdxT` size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
 - static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` size, double pos_↵, sigma=default_sigma_pos)
 - static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos_↵, beta=default_beta_pos)
 - static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=default_mean_↵, double kappa=default_intensity_kappa)
 - static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min_sigma, double max_↵, sigma, double alpha=default_alpha_sigma)
 - static void `set_rng_seed` (`RngSeedT` seed)
 - static `ParallelRngManagerT` & `get_rng_manager` ()
 - static `ParallelRngGeneratorT` & `get_rng_generator` ()
 - static void `check_size` (const `ImageSizeT` &size_)
- Check the size argument for the model.*

Static Public Attributes

- static const std::string `name`
- static const `StringVecT` `prior_types`
- static const std::string `DefaultPriorType` = "Normal"
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6

- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) = 1
- static const [ImageCoordT](#) [global_min_size](#) = 3
- static const [ImageCoordT](#) [global_max_size](#) = 512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [set_mcmc_num_phases](#) ([IdxT](#) num_phases)

Protected Attributes

- double [psf_sigma](#)
- CompositeDist [prior](#)
- [IdxT](#) [num_params](#)
- [IdxT](#) [num_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta_x](#) =0
- double [eta_l](#) =0
- double [eta_bg](#) =0
- [IdxT](#) [num_phases](#)
- double [sigma_scale](#)

8.6.1 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.

8.6.2 Member Typedef Documentation

8.6.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

8.6.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

8.6.2.3 `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.

8.6.2.4 `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.

8.6.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.6.2.6 `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.

8.6.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>` [inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.6.2.8 `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.

8.6.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.6.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.6.2.11 `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.

8.6.2.12 `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.

8.6.2.13 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.6.2.14 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.6.2.15 `using mappel::Gauss1DModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 49 of file Gauss1DModel.h.

8.6.3 Constructor & Destructor Documentation

8.6.3.1 `mappel::Gauss1DMAP::Gauss1DMAP (arma::Col< ImageCoordT > size, VecT psf_sigma, const std::string & prior_type = DefaultPriorType)`

Definition at line 11 of file Gauss1DMAP.cpp.

8.6.3.2 `mappel::Gauss1DMAP::Gauss1DMAP (ImageSizeT size, double psf_sigma, const std::string & prior_type = DefaultPriorType)`

Definition at line 15 of file Gauss1DMAP.cpp.

8.6.3.3 `mappel::Gauss1DMAP::Gauss1DMAP (ImageSizeT size, double psf_sigma, CompositeDist && prior)`

Definition at line 19 of file Gauss1DMAP.cpp.

8.6.3.4 `mappel::Gauss1DMAP::Gauss1DMAP (ImageSizeT size, double psf_sigma, const CompositeDist & prior)`

Definition at line 27 of file Gauss1DMAP.cpp.

8.6.3.5 `mappel::Gauss1DMAP::Gauss1DMAP (const Gauss1DMAP & o)`

Definition at line 35 of file Gauss1DMAP.cpp.

8.6.3.6 `mappel::Gauss1DMAP::Gauss1DMAP (Gauss1DMAP && o)`

Definition at line 43 of file Gauss1DMAP.cpp.

8.6.4 Member Function Documentation

8.6.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const`
[*inherited*]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.6.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [*inherited*]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.6.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.6.4.4 `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`.

8.6.4.5 `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`.

8.6.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.6.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.6.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.6.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.6.4.10 `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()`.

8.6.4.11 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` [inline],[inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.6.4.13 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.6.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` [inline],[inherited]

Definition at line 224 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.6.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` [inline],[inherited]

Definition at line 108 of file ImageFormat1DBase.h.

8.6.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.6.4.17 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.6.4.18 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.6.4.19 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`, `[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.6.4.20 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`, `[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.6.4.21 `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().

8.6.4.22 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline]`, `[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.6.4.25 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().

8.6.4.26 double mappel::Gauss1DModel::get_psf_sigma (IdxT idx) const [inherited]

Definition at line 131 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::psf_sigma.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.6.4.29 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().

8.6.4.30 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT idx) const` `[inherited]`

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

8.6.4.31 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & stack) const` `[inline], [inherited]`

Definition at line 101 of file ImageFormat1DBase.h.

8.6.4.32 `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().

8.6.4.33 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const` `[inline], [inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.6.4.34 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const` `[inline], [inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.6.4.35 `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.

8.6.4.36 `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.

8.6.4.37 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()`.

8.6.4.38 CompositeDist `mappel::Gauss1DModel::make_default_prior_beta_position (IdxT size)` [static], [inherited]

Definition at line 80 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_beta()`.

Referenced by `mappel::Gauss1DModel::make_default_prior()`.

8.6.4.39 CompositeDist `mappel::Gauss1DModel::make_default_prior_normal_position (IdxT size)` [static], [inherited]

Definition at line 90 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_normal()`.

Referenced by `mappel::Gauss1DModel::make_default_prior()`.

8.6.4.40 ImageFormat1DBase::ImageT `ImageFormat1DBase::make_image () const` [inline], [inherited]

Definition at line 87 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

8.6.4.41 ImageFormat1DBase::ImageStackT `ImageFormat1DBase::make_image_stack (ImageCoordT n) const` [inline], [inherited]

Definition at line 94 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

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

Definition at line 164 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

8.6.4.43 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const`
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.6.4.44 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline], [inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.6.4.45 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.6.4.46 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline], [inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.6.4.47 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.6.4.48 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline], [inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.6.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.6.4.50 `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()`.

8.6.4.51 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.6.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

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

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

Definition at line 85 of file PointEmitterModel.cpp.

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

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.6.4.55 `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()`.

8.6.4.56 `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()`.

8.6.4.57 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=().

8.6.4.58 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=().

8.6.4.59 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().

8.6.4.60 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.

8.6.4.61 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.

8.6.4.62 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.

8.6.4.63 `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`.

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

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.6.4.65 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const` `[inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.6.4.66 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)` `[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.6.4.67 `void mappel::MCMCAdaptor1D::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)` `[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`.

8.6.4.68 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)` `[inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior, and mappel::rng_manager.

8.6.4.70 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_pixel_mean_bg, mappel::MCMCAdaptor1D::eta_bg, mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_lbound(), mappel::PointEmitterModel::get_ubound(), and mappel::MCMCAdaptorBase::sigma_scale.

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

8.6.4.71 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.6.4.73 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` [inline],[inherited]

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.6.4.74 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` [inline],[inherited]

Definition at line 220 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::set_hyperparams(), and mappel::Gauss2DsModel::set_hyperparams().

8.6.4.75 `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.

8.6.4.76 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1)` `[inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.6.4.77 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.6.4.78 `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()`.

8.6.4.79 `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`.

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

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 158 of file PointEmitterModel.cpp.

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

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

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

Definition at line 149 of file PointEmitterModel.cpp.

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

8.6.4.83 `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()`.

8.6.4.84 `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()`.

8.6.4.85 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` `[static]`, `[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.6.4.86 `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()`.

8.6.4.87 `void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & sz)` `[inline]`, `[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

8.6.4.88 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.6.5 Member Data Documentation

8.6.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.6.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.6.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.6.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.6.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.6.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.6.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.6.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 53 of file Gauss1DModel.h.

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

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

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

Definition at line 24 of file PoissonNoise1DObjective.h.

8.6.5.12 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.6.5.13 `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.6.5.14 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.6.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.6.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.6.5.17 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

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

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

8.6.5.18 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

8.6.5.19 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

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

8.6.5.20 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.6.5.22 `const std::string mappel::Gauss1DMP::name` `[static]`

Definition at line 34 of file `Gauss1DMP.h`.

8.6.5.23 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static], [inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::get_stats().

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::get_num_params(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::make_param(), mappel::PointEmitterModel::make_param_mat(), mappel::PointEmitterModel::make_param_mat_stack(), mappel::PointEmitterModel::make_param_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.6.5.26 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected], [inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_num_phases(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor1Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor2Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor2D::sample_mcmc_candidate(), mappel::MCMCAdaptor1D::sample_mcmc_candidate(), and mappel::MCMCAdaptorBase::set_mcmc_num_phases().

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

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get_hyperparam_index(), mappel::PointEmitterModel::get_hyperparam_names(), mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_hyperparams(), mappel::Gauss1DsModel::get_max_sigma(), mappel::Gauss1DsModel::get_min_sigma(), mappel::PointEmitterModel::get_param_names(), mappel::PointEmitterModel::get_prior(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::has_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename_hyperparam(), mappel::PointEmitterModel::sample_prior(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_hyperparam_names(), mappel::PointEmitterModel::set_hyperparam_value(), mappel::PointEmitterModel::set_hyperparams(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::PointEmitterModel::set_param_names(), mappel::PointEmitterModel::set_prior(), and mappel::PointEmitterModel::set_ubound().

8.6.5.28 `const StringVecT mappel::Gauss1DModel::prior_types` `[static],[inherited]`

Initial value:

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

Definition at line 52 of file Gauss1DModel.h.

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

8.6.5.29 `double mappel::Gauss1DModel::psf_sigma` `[protected],[inherited]`

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 90 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::get_psf_sigma()`, `mappel::Gauss1DModel::operator=()`, `mappel::Gauss1DModel::pixel_grad2()`, `mappel::Gauss1DModel::pixel_hess()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

8.6.5.30 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected],[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.6.5.31 `ImageSizeT mappel::ImageFormat1DBase::size` `[protected],[inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

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

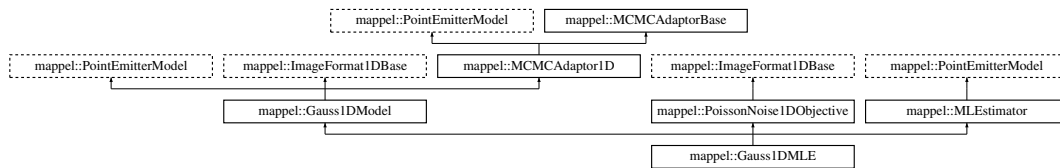
- [Gauss1DMap.h](#)
- [Gauss1DMap.cpp](#)

8.7 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)
- [ParamT sample_prior](#) ()
- 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)
- void [set_intensity_mcmc_sampling](#) (double eta_l=-1)
- void [set_background_mcmc_sampling](#) (double eta_bg=-1)
- void [set_mcmc_sigma_scale](#) (double scale)
- double [get_mcmc_sigma_scale](#) () const
- [IdxT get_mcmc_num_phases](#) () const

Static Public Member Functions

- static CompositeDist [make_default_prior](#) (IdxT size, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) (IdxT size)
- static CompositeDist [make_default_prior_normal_position](#) (IdxT size)
- static CompositeDist [make_prior_beta_position](#) (IdxT size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
- static CompositeDist [make_prior_normal_position](#) (IdxT size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) (IdxT size, double pos_↵ sigma=default_sigma_pos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_↵ beta=default_beta_pos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=default_mean_↵ l, double kappa=default_intensity_kappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵ sigma, double alpha=default_alpha_sigma)
- static void [set_rng_seed](#) (RngSeedT seed)
- static ParallelRngManagerT & [get_rng_manager](#) ()
- static ParallelRngGeneratorT & [get_rng_generator](#) ()
- static void [check_size](#) (const ImageSizeT &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [name](#)
- static const StringVecT [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const ImageCoordT [num_dim](#) = 1
- static const ImageCoordT [global_min_size](#) = 3
- static const ImageCoordT [global_max_size](#) = 512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [set_mcmc_num_phases](#) (IdxT num_phases)

Protected Attributes

- double [psf_sigma](#)
- CompositeDist [prior](#)
- [IdxT](#) [num_params](#)
- [IdxT](#) [num_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta_x](#) =0
- double [eta_l](#) =0
- double [eta_bg](#) =0
- [IdxT](#) [num_phases](#)
- double [sigma_scale](#)

8.7.1 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.

8.7.2 Member Typedef Documentation

8.7.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [\[inherited\]](#)

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

8.7.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` [\[inherited\]](#)

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

8.7.2.3 `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.

8.7.2.4 `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.

8.7.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.7.2.6 `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.

8.7.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
[inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.7.2.8 `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.

8.7.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
[inherited]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.7.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.7.2.11 `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.

8.7.2.12 `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.

8.7.2.13 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.7.2.14 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.7.2.15 `using mappel::Gauss1DModel::StencilVecT = std::vector<Stencil>` [inherited]

Definition at line 49 of file Gauss1DModel.h.

8.7.3 Constructor & Destructor Documentation

8.7.3.1 `mappel::Gauss1DMLE::Gauss1DMLE (arma::Col< ImageCoordT > size, VecT psf_sigma, const std::string & prior_type = DefaultPriorType)`

Definition at line 11 of file Gauss1DMLE.cpp.

8.7.3.2 `mappel::Gauss1DMLE::Gauss1DMLE (ImageSizeT size, double psf_sigma, const std::string & prior_type = DefaultPriorType)`

Definition at line 15 of file Gauss1DMLE.cpp.

8.7.3.3 `mappel::Gauss1DMLE::Gauss1DMLE (ImageSizeT size, double psf_sigma, CompositeDist && prior)`

Definition at line 19 of file Gauss1DMLE.cpp.

8.7.3.4 mappel::Gauss1DMLE::Gauss1DMLE (ImageSizeT size, double psf_sigma, const CompositeDist & prior)

Definition at line 27 of file Gauss1DMLE.cpp.

8.7.3.5 mappel::Gauss1DMLE::Gauss1DMLE (const Gauss1DMLE & o)

Definition at line 35 of file Gauss1DMLE.cpp.

8.7.3.6 mappel::Gauss1DMLE::Gauss1DMLE (Gauss1DMLE && o)

Definition at line 43 of file Gauss1DMLE.cpp.

8.7.4 Member Function Documentation

8.7.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

8.7.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

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

8.7.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::make_param_stack().

8.7.4.4 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.

8.7.4.5 `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.

8.7.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.7.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.7.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.7.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.7.4.10 `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().

8.7.4.11 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.7.4.13 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.7.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.7.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline]`,`[inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

8.7.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.7.4.17 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.7.4.18 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

8.7.4.19 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

8.7.4.20 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.7.4.21 `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()`.

8.7.4.22 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 200 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

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

Definition at line 204 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.7.4.25 `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()`.

8.7.4.26 `double mappel::Gauss1DModel::get_psf_sigma (IdxT idx) const` `[inherited]`

Definition at line 131 of file Gauss1DModel.cpp.

References `mappel::Gauss1DModel::psf_sigma`.

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

Definition at line 120 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

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

Definition at line 115 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.7.4.29 `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()`.

8.7.4.30 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT idx) const` `[inherited]`

Definition at line 20 of file ImageFormat1DBase.cpp.

References `mappel::ImageFormat1DBase::size`.

8.7.4.31 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & stack) const` `[inline],[inherited]`

Definition at line 101 of file ImageFormat1DBase.h.

8.7.4.32 `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()`.

8.7.4.33 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const` `[inline]`,
`[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.7.4.34 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const` `[inline]`, `[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.7.4.35 `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.

8.7.4.36 `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.

8.7.4.37 `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().

8.7.4.38 `CompositeDist mappel::Gauss1DModel::make_default_prior_beta_position (IdxT size)` `[static]`,
`[inherited]`

Definition at line 80 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default_pixel_mean_bg, mappel::PointEmitterModel::make_prior_component_intensity(), and mappel::PointEmitterModel::make_prior_component_position_beta().

Referenced by mappel::Gauss1DModel::make_default_prior().

8.7.4.39 CompositeDist `mappel::Gauss1DModel::make_default_prior_normal_position (IdxT size) [static], [inherited]`

Definition at line 90 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_`
`_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_normal()`.

Referenced by `mappel::Gauss1DModel::make_default_prior()`.

8.7.4.40 ImageFormat1DBase::ImageT `ImageFormat1DBase::make_image () const [inline], [inherited]`

Definition at line 87 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

8.7.4.41 ImageFormat1DBase::ImageStackT `ImageFormat1DBase::make_image_stack (ImageCoordT n) const [inline], [inherited]`

Definition at line 94 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

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

Definition at line 164 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

8.7.4.43 template<class FillT > PointEmitterModel::ParamT `mappel::PointEmitterModel::make_param (FillT fill) const [inherited]`

Definition at line 181 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.7.4.44 MatT `mappel::PointEmitterModel::make_param_mat () const [inline], [inherited]`

Definition at line 172 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.7.4.45 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.7.4.46 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` [inline], [inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.7.4.47 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
[inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.7.4.48 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` [inline],
[inherited]

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta↵_stack().

8.7.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` [inherited]

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.7.4.50 `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().

8.7.4.51 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa) [static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.7.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos) [static],[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

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

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

Definition at line 85 of file PointEmitterModel.cpp.

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

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.7.4.55 **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().

8.7.4.56 **Gauss1DModel::Stencil** mappel::Gauss1DModel::make_stencil (*const ParamT & theta*, *bool compute_derivatives = true*) *const* [inline], [inherited]

Make a new Model::Stencil object at theta.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular theta (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the [mappel::methods](#) namespace accept a *const Stencil* reference in place of the model parameter.

Throws [mappel::ModelBoundsError](#) if not model.theta_in_bounds(theta).

If derivatives will not be computed with this stencil set compute_derivatives=false

Parameters

<i>theta</i>	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 116 of file Gauss1DModel.h.

References mappel::Gauss1DModel::Stencil::Stencil(), and mappel::PointEmitterModel::theta_in_bounds().

Referenced by mappel::Gauss1DModel::initial_theta_estimate().

8.7.4.57 **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=().

8.7.4.58 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=().

8.7.4.59 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().

8.7.4.60 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.

8.7.4.61 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.

8.7.4.62 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.

8.7.4.63 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.

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

Definition at line 276 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

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

8.7.4.65 PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & *theta*)
const [inherited]

Definition at line 317 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::make_param_stack(), and mappel::PointEmitterModel::reflected_theta().

8.7.4.66 void mappel::PointEmitterModel::rename_hyperparam (const std::string & *old_name*, const std::string & *new_name*)
[inline], [inherited]

Definition at line 244 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.7.4.67 void mappel::MCMCAdaptor1D::sample_mcmc_candidate (IdxT *sample_index*, ParamT & *candidate*, double *step_scale* = 1.0) [inherited]

Definition at line 108 of file MCMCAdaptor1D.cpp.

References mappel::MCMCAdaptor1D::eta_bg, mappel::MCMCAdaptor1D::eta_I, mappel::MCMCAdaptor1D::eta_x, mappel::MCMCAdaptorBase::num_phases, and mappel::rng_manager.

8.7.4.68 template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & *rng*)
[inherited]

Definition at line 264 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.7.4.69 PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () [inline], [inherited]

Definition at line 268 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior, and mappel::rng_manager.

8.7.4.70 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` `[inherited]`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.7.4.71 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.7.4.73 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` `[inline]`,
`[inherited]`

Definition at line 240 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.7.4.74 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` `[inline]`, `[inherited]`

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

8.7.4.75 `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.

8.7.4.76 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_l = -1)` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_l`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.7.4.77 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.7.4.78 `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()`.

8.7.4.79 `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`.

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

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 158 of file PointEmitterModel.cpp.

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

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

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

Definition at line 149 of file PointEmitterModel.cpp.

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

8.7.4.83 `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()`.

8.7.4.84 `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()`.

8.7.4.85 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static], [inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.7.4.86 `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()`.

8.7.4.87 `void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & sz) [inline], [inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

8.7.4.88 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.7.5 Member Data Documentation

8.7.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.7.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.7.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.7.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.7.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.7.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.7.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.7.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 53 of file Gauss1DModel.h.

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

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

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

Definition at line 24 of file PoissonNoise1DObjective.h.

8.7.5.12 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.7.5.13 `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.7.5.14 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.7.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.7.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.7.5.17 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

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

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

8.7.5.18 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

8.7.5.19 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

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

8.7.5.20 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

Definition at line 34 of file `Gauss1DMLE.h`.

8.7.5.23 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static], [inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::get_stats().

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::get_num_params(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::make_param(), mappel::PointEmitterModel::make_param_mat(), mappel::PointEmitterModel::make_param_mat_stack(), mappel::PointEmitterModel::make_param_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.7.5.26 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected], [inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_num_phases(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor2Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor2D::sample_mcmc_candidate(), mappel::MCMCAdaptor1Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor1D::sample_mcmc_candidate(), and mappel::MCMCAdaptorBase::set_mcmc_num_phases().

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

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get_hyperparam_index(), mappel::PointEmitterModel::get_hyperparam_names(), mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_hyperparams(), mappel::Gauss1DsModel::get_max_sigma(), mappel::Gauss1DsModel::get_min_sigma(), mappel::PointEmitterModel::get_param_names(), mappel::PointEmitterModel::get_prior(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::has_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename_hyperparam(), mappel::PointEmitterModel::sample_prior(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_hyperparam_names(), mappel::PointEmitterModel::set_hyperparam_value(), mappel::PointEmitterModel::set_hyperparams(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::PointEmitterModel::set_param_names(), mappel::PointEmitterModel::set_prior(), and mappel::PointEmitterModel::set_ubound().

8.7.5.28 `const StringVecT mappel::Gauss1DModel::prior_types` `[static],[inherited]`

Initial value:

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

Definition at line 52 of file Gauss1DModel.h.

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

8.7.5.29 `double mappel::Gauss1DModel::psf_sigma` `[protected],[inherited]`

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 90 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::get_psf_sigma()`, `mappel::Gauss1DModel::operator=()`, `mappel::Gauss1DModel::pixel_grad2()`, `mappel::Gauss1DModel::pixel_hess()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

8.7.5.30 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected],[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.7.5.31 `ImageSizeT mappel::ImageFormat1DBase::size` `[protected],[inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

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

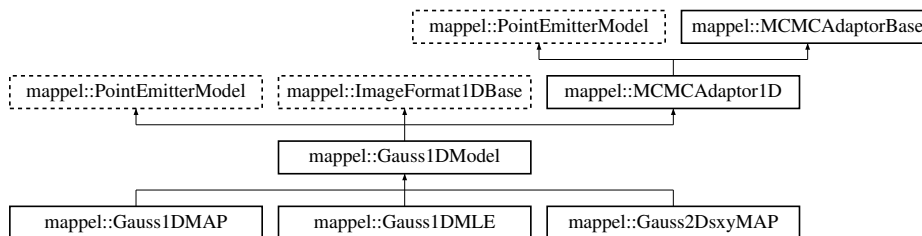
- [Gauss1DMLE.h](#)
- [Gauss1DMLE.cpp](#)

8.8 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)
- [ParamT sample_prior](#) ()
- 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)
- void [set_intensity_mcmc_sampling](#) (double eta_l=-1)
- void [set_background_mcmc_sampling](#) (double eta_bg=-1)
- void [set_mcmc_sigma_scale](#) (double scale)
- double [get_mcmc_sigma_scale](#) () const
- IdxT [get_mcmc_num_phases](#) () const

Static Public Member Functions

- static CompositeDist [make_default_prior](#) (IdxT size, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) (IdxT size)
- static CompositeDist [make_default_prior_normal_position](#) (IdxT size)
- static CompositeDist [make_prior_beta_position](#) (IdxT size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
- static CompositeDist [make_prior_normal_position](#) (IdxT size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) (IdxT size, double pos_↵ sigma=default_sigma_pos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_↵ beta=default_beta_pos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=default_mean_↵ l, double kappa=default_intensity_kappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵ sigma, double alpha=default_alpha_sigma)
- static void [set_rng_seed](#) (RngSeedT seed)
- static ParallelRngManagerT & [get_rng_manager](#) ()
- static ParallelRngGeneratorT & [get_rng_generator](#) ()
- static void [check_size](#) (const ImageSizeT &size_)

Check the size argument for the model.

Static Public Attributes

- static const StringVecT [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const ImageCoordT [num_dim](#) = 1
- static const ImageCoordT [global_min_size](#) = 3
- static const ImageCoordT [global_max_size](#) = 512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5

Protected Member Functions

- [Gauss1DModel](#) ([IdxT](#) size, double [psf_sigma](#))
- [Gauss1DModel](#) (const [Gauss1DModel](#) &o)
- [Gauss1DModel](#) ([Gauss1DModel](#) &&o)
- [Gauss1DModel](#) & operator= (const [Gauss1DModel](#) &o)
- [Gauss1DModel](#) & operator= ([Gauss1DModel](#) &&o)
- void [set_mcmc_num_phases](#) ([IdxT](#) num_phases)

Protected Attributes

- double [psf_sigma](#)
- CompositeDist [prior](#)
- [IdxT](#) num_params
- [IdxT](#) num_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double [eta_x](#) =0
- double [eta_l](#) =0
- double [eta_bg](#) =0
- [IdxT](#) num_phases
- double [sigma_scale](#)

8.8.1 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](#).

8.8.2 Member Typedef Documentation

8.8.2.1 using [mappel::ImageFormat1DBase::ImageCoordT](#) = [uint32_t](#) [inherited]

Image size coordinate storage type

Definition at line 25 of file [ImageFormat1DBase.h](#).

8.8.2.2 using [mappel::ImageFormat1DBase::ImagePixelT](#) = [double](#) [inherited]

Image pixel storage type

Definition at line 26 of file [ImageFormat1DBase.h](#).

8.8.2.3 `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.

8.8.2.4 `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.

8.8.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.8.2.6 `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.

8.8.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
[inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.8.2.8 `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.

8.8.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
[inherited]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.8.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.8.2.11 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.8.2.12 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.8.2.13 `using mappel::Gauss1DModel::StencilVecT = std::vector<Stencil>`

Definition at line 49 of file Gauss1DModel.h.

8.8.3 Constructor & Destructor Documentation

8.8.3.1 `mappel::Gauss1DModel::Gauss1DModel (IdxT size, double psf_sigma)` [protected]

Definition at line 12 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::check_psf_sigma()`.

8.8.3.2 `mappel::Gauss1DModel::Gauss1DModel (const Gauss1DModel & o)` [protected]

Definition at line 20 of file Gauss1DModel.cpp.

8.8.3.3 `mappel::Gauss1DModel::Gauss1DModel (Gauss1DModel && o)` [protected]

Definition at line 26 of file Gauss1DModel.cpp.

8.8.4 Member Function Documentation

8.8.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const`
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.8.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.8.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.8.4.4 `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`.

8.8.4.5 `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`.

8.8.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.8.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.8.4.8 void mappel::PointEmitterModel::check_psf_sigma (double *psf_sigma*) const [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.8.4.9 void mappel::PointEmitterModel::check_psf_sigma (const VecT & *psf_sigma*) const [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.8.4.10 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().

8.8.4.11 int mappel::PointEmitterModel::get_hyperparam_index (const std::string & *name*) const [inline],[inherited]

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.8.4.13 double mappel::PointEmitterModel::get_hyperparam_value (const std::string & *name*) const [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.8.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const [inline], [inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.8.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const [inline], [inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

8.8.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const [inline], [inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.8.4.17 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const [inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.8.4.18 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const [inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.8.4.19 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const [inline], [inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.8.4.20 `IdxT mappel::PointEmitterModel::get_num_params () const [inline], [inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.8.4.21 `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().

8.8.4.22 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline]`,`[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel↵
 ::update_internal_1Dsum_estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.8.4.25 `double mappel::Gauss1DModel::get_psf_sigma () const` `[inline]`

Definition at line 127 of file Gauss1DModel.h.

References psf_sigma.

Referenced by get_stats().

8.8.4.26 `double mappel::Gauss1DModel::get_psf_sigma (IdxT idx) const`

Definition at line 131 of file Gauss1DModel.cpp.

References psf_sigma.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.8.4.29 **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().

8.8.4.30 **ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT idx)** const [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

8.8.4.31 **ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & stack)** const [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

8.8.4.32 **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().

8.8.4.33 **const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound ()** const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.8.4.34 **bool mappel::PointEmitterModel::has_hyperparam (const std::string & name)** const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.8.4.35 `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`.

8.8.4.36 `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`.

8.8.4.37 `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()`.

8.8.4.38 `CompositeDist mappel::Gauss1DModel::make_default_prior_beta_position (IdxT size)` `[static]`

Definition at line 80 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_beta()`.

Referenced by `make_default_prior()`.

8.8.4.39 `CompositeDist mappel::Gauss1DModel::make_default_prior_normal_position (IdxT size)` `[static]`

Definition at line 90 of file Gauss1DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, and `mappel::PointEmitterModel::make_prior_component_position_normal()`.

Referenced by `make_default_prior()`.

8.8.4.40 `ImageFormat1DBase::ImageT ImageFormat1DBase::make_image () const` `[inline],[inherited]`

Definition at line 87 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

8.8.4.41 `ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack (ImageCoordT n) const` `[inline],[inherited]`

Definition at line 94 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

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

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.8.4.43 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const [inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.8.4.44 MatT mappel::PointEmitterModel::make_param_mat () const [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.8.4.45 template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.8.4.46 CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.8.4.47 template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.8.4.48 PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const [inline],[inherited]

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.8.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.8.4.50 `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().

8.8.4.51 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.8.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` `[static],[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_beta_position(), make_prior_beta_position(), and mappel::Gauss2DsModel::make_prior_beta_position().

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

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

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

Definition at line 104 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.8.4.55 `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()`.

8.8.4.56 `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()`.

8.8.4.57 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=().

8.8.4.58 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.

8.8.4.59 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().

8.8.4.60 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.

8.8.4.61 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.

8.8.4.62 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.

8.8.4.63 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.

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

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.8.4.65 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const`
`[inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.8.4.66 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)`
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.8.4.67 `void mappel::MCMCAdaptor1D::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)`
`[inherited]`

Definition at line 108 of file `MCMCAdaptor1D.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

8.8.4.68 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.8.4.70 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.8.4.71 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`
[inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.8.4.73 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` [inline],
[inherited]

Definition at line 240 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.8.4.74 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` [inline], [inherited]

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

8.8.4.75 `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.

8.8.4.76 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1)` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.8.4.77 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.8.4.78 `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()`.

8.8.4.79 `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`.

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

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 158 of file PointEmitterModel.cpp.

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

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

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

Definition at line 149 of file PointEmitterModel.cpp.

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

8.8.4.83 `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()`.

8.8.4.84 `void mappel::Gauss1DModel::set_psf_sigma (const VecT & new_psf_sigma)` `[inline]`

Definition at line 131 of file Gauss1DModel.h.

References `set_psf_sigma()`.

8.8.4.85 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` `[static]`, `[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.8.4.86 `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()`.

8.8.4.87 `void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & sz)` `[inline]`, `[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

8.8.4.88 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.8.5 Member Data Documentation

8.8.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static]`, `[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.8.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static]`, `[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.8.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static]`, `[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.8.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static]`, `[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.8.5.5 `const double mappel::PointEmitterModel::default_max_I = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.8.5.6 `const double mappel::PointEmitterModel::default_mean_I = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.8.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.8.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 53 of file Gauss1DModel.h.

Referenced by `operator=()`.

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

8.8.5.11 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.8.5.12 `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.8.5.13 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.8.5.14 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

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

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.8.5.15 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.8.5.16 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static]`, `[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

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

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

8.8.5.17 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static]`, `[inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

8.8.5.18 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static]`, `[inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

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

8.8.5.19 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static]`, `[inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.8.5.21 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static], [inherited]`

Number of image dimensions.

Definition at line 38 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

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

Definition at line 147 of file `PointEmitterModel.h`.

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

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

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `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()`.

8.8.5.24 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected], [inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

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

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

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

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

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

Initial value:

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

Definition at line 52 of file Gauss1DModel.h.

Referenced by operator=().

8.8.5.27 `double mappel::Gauss1DModel::psf_sigma` `[protected]`

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 90 of file Gauss1DModel.h.

Referenced by mappel::Gauss1DModel::Stencil::compute_derivatives(), get_psf_sigma(), operator=(), pixel_grad2(), pixel_hess(), pixel_hess_update(), set_psf_sigma(), and mappel::Gauss1DModel::Stencil::Stencil().

8.8.5.28 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected],[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_sigma_scale(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling(), and mappel::MCMCAdaptorBase::set_mcmc_sigma_scale().

8.8.5.29 `ImageSizeT mappel::ImageFormat1DBase::size` `[protected],[inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by mappel::ImageFormat1DBase::check_image_shape(), mappel::Gauss1DsModel::Stencil::compute_derivatives(), mappel::Gauss1DModel::Stencil::compute_derivatives(), mappel::ImageFormat1DBase::get_num_pixels(), mappel::ImageFormat1DBase::get_size(), initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::ImageFormat1DBase::make_image(), mappel::ImageFormat1DBase::make_image_stack(), mappel::ImageFormat1DBase::set_size(), mappel::Gauss1DsModel::Stencil::Stencil(), and mappel::Gauss1DModel::Stencil::Stencil().

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

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

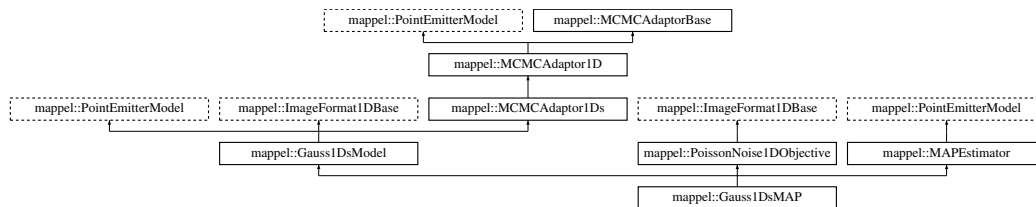
- [Gauss1DModel.h](#)
- [Gauss1DModel.cpp](#)

8.9 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=DefaultPriorType)
- [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)`
- `ParamT sample_prior ()`
- `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)`
- `void set_intensity_mcmc_sampling (double eta_l=-1)`
- `void set_background_mcmc_sampling (double eta_bg=-1)`
- `void set_mcmc_sigma_scale (double scale)`
- `double get_mcmc_sigma_scale () const`
- `IdxT get_mcmc_num_phases () const`

Static Public Member Functions

- static CompositeDist [make_default_prior](#) (IdxT size, double min_sigma, double max_sigma, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) (IdxT size, double min_sigma, double max_sigma)
- static CompositeDist [make_default_prior_normal_position](#) (IdxT size, double min_sigma, double max_sigma)
- static CompositeDist [make_prior_beta_position](#) (IdxT size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma)
- static CompositeDist [make_prior_normal_position](#) (IdxT size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) (IdxT size, double pos_←, sigma=default_sigma_pos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_←, beta=default_beta_pos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=default_mean_←, l, double kappa=default_intensity_kappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_←, sigma, double alpha=default_alpha_sigma)
- static void [set_rng_seed](#) (RngSeedT seed)
- static ParallelRngManagerT & [get_rng_manager](#) ()
- static ParallelRngGeneratorT & [get_rng_generator](#) ()
- static void [check_size](#) (const ImageSizeT &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) = 1
- static const [ImageCoordT](#) [global_min_size](#) = 3
- static const [ImageCoordT](#) [global_max_size](#) = 512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [set_mcmc_num_phases](#) (IdxT num_phases)

Protected Attributes

- CompositeDist [prior](#)
- [IdxT](#) num_params
- [IdxT](#) num_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double [eta_sigma](#) = -1
- double [eta_x](#) = 0
- double [eta_l](#) = 0
- double [eta_bg](#) = 0
- [IdxT](#) num_phases
- double [sigma_scale](#)

8.9.1 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.

8.9.2 Member Typedef Documentation

8.9.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

8.9.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

8.9.2.3 `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.

8.9.2.4 `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.

8.9.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.9.2.6 `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.

8.9.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
[inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.9.2.8 `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.

8.9.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
[inherited]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.9.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.9.2.11 `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.

8.9.2.12 `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.

8.9.2.13 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.9.2.14 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.9.2.15 `using mappel::Gauss1DsModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 47 of file Gauss1DsModel.h.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 `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.

8.9.3.2 `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.

8.9.3.3 `mappel::Gauss1DsMAP::Gauss1DsMAP (ImageSizeT size, CompositeDist && prior)`

Definition at line 20 of file Gauss1DsMAP.cpp.

8.9.3.4 mappel::Gauss1DsMAP::Gauss1DsMAP (ImageSizeT *size*, const CompositeDist & *prior*)

Definition at line 28 of file Gauss1DsMAP.cpp.

8.9.3.5 mappel::Gauss1DsMAP::Gauss1DsMAP (const Gauss1DsMAP & *o*)

Definition at line 36 of file Gauss1DsMAP.cpp.

8.9.3.6 mappel::Gauss1DsMAP::Gauss1DsMAP (Gauss1DsMAP && *o*)

Definition at line 44 of file Gauss1DsMAP.cpp.

8.9.4 Member Function Documentation

8.9.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

8.9.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

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

8.9.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::make_param_stack().

8.9.4.4 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.

8.9.4.5 `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.

8.9.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` `[inherited]`

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.9.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` `[inherited]`

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.9.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` `[inherited]`

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.9.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.9.4.10 `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().

8.9.4.11 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.9.4.13 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.9.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.9.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline]`,`[inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

8.9.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.9.4.17 `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().

8.9.4.18 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

8.9.4.19 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

8.9.4.20 `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()`.

8.9.4.21 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

8.9.4.22 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.9.4.23 `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()`.

8.9.4.24 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.9.4.29 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().

8.9.4.30 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT idx) const [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

8.9.4.31 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

8.9.4.32 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().

8.9.4.33 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline], [inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.9.4.34 bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline], [inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.9.4.35 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().

8.9.4.36 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.

8.9.4.37 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().

8.9.4.38 `CompositeDist mappel::Gauss1DsModel::make_default_prior_beta_position (IdxT size, double min_sigma, double max_sigma) [static], [inherited]`

Definition at line 72 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_←_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::PointEmitterModel_←::make_prior_component_sigma()`.

Referenced by `mappel::Gauss1DsModel::make_default_prior()`.

8.9.4.39 `CompositeDist mappel::Gauss1DsModel::make_default_prior_normal_position (IdxT size, double min_sigma, double max_sigma) [static], [inherited]`

Definition at line 83 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_←_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::PointEmitterModel_←::make_prior_component_sigma()`.

Referenced by `mappel::Gauss1DsModel::make_default_prior()`.

8.9.4.40 `ImageFormat1DBase::ImageT ImageFormat1DBase::make_image () const [inline], [inherited]`

Definition at line 87 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

8.9.4.41 `ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack (ImageCoordT n) const [inline], [inherited]`

Definition at line 94 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

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

Definition at line 164 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::_←Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel_←::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_←update()`.

8.9.4.43 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const [inherited]`

Definition at line 181 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.9.4.44 `MatT mappel::PointEmitterModel::make_param_mat () const [inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.9.4.45 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const [inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.9.4.46 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const [inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.9.4.47 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const [inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.9.4.48 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const [inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.9.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const [inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.9.4.50 `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().

8.9.4.51 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_I, double kappa = default_intensity_kappa)` [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.9.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

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

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

Definition at line 85 of file PointEmitterModel.cpp.

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

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.9.4.55 `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()`.

8.9.4.56 `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()`.

8.9.4.57 `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=()`.

8.9.4.58 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=().

8.9.4.59 void mappel::Gauss1DsModel::pixel_grad (IdxT i, const Stencil & s, ParamT & pgrad) const [inline], [inherited]

Definition at line 140 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::I(), and mappel::Gauss1DsModel::Stencil::X.

Referenced by mappel::Gauss1DsModel::pixel_hess_update().

8.9.4.60 void mappel::Gauss1DsModel::pixel_grad2 (IdxT i, const Stencil & s, ParamT & pgrad2) const [inline], [inherited]

Definition at line 150 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::I(), and mappel::Gauss1DsModel::Stencil::sigma().

8.9.4.61 void mappel::Gauss1DsModel::pixel_hess (IdxT i, const Stencil & s, MatT & hess) const [inline], [inherited]

Definition at line 160 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::DXSX, mappel::Gauss1DsModel::Stencil::I(), and mappel::Gauss1DsModel::Stencil::sigma().

8.9.4.62 void mappel::Gauss1DsModel::pixel_hess_update (IdxT i, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess) const [inherited]

Definition at line 211 of file Gauss1DsModel.cpp.

References mappel::Gauss1DsModel::Stencil::DX, mappel::Gauss1DsModel::Stencil::DXS, mappel::Gauss1DsModel::Stencil::DXS2, mappel::Gauss1DsModel::Stencil::DXSX, mappel::Gauss1DsModel::Stencil::I(), mappel::PointEmitterModel::make_param(), mappel::Gauss1DsModel::pixel_grad(), and mappel::Gauss1DsModel::Stencil::sigma().

8.9.4.63 double mappel::Gauss1DsModel::pixel_model_value (IdxT i, const Stencil & s) const [inline], [inherited]

Definition at line 134 of file Gauss1DsModel.h.

References mappel::Gauss1DsModel::Stencil::bg(), mappel::Gauss1DsModel::Stencil::I(), and mappel::Gauss1DsModel::Stencil::X.

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

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.9.4.65 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const`
`[inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.9.4.66 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)`
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.9.4.67 `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)`
`[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`.

8.9.4.68 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.9.4.70 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.9.4.71 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.9.4.73 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` [inline], [inherited]

Definition at line 240 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.9.4.74 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` [inline], [inherited]

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

8.9.4.75 `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.

8.9.4.76 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1)` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.9.4.77 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.9.4.78 `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()`.

8.9.4.79 `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()`.

8.9.4.80 `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()`.

8.9.4.81 `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`.

8.9.4.82 `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()`.

8.9.4.83 `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()`.

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

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 158 of file PointEmitterModel.cpp.

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

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

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

Definition at line 149 of file PointEmitterModel.cpp.

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

8.9.4.87 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` [static],[inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.9.4.88 `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()`.

8.9.4.89 `void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & sz) [inline], [inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

8.9.4.90 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.9.5 Member Data Documentation

8.9.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.9.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2 [static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.9.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.9.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.9.5.5 `const double mappel::PointEmitterModel::default_max_I = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.9.5.6 `const double mappel::PointEmitterModel::default_mean_I = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.9.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.9.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 51 of file Gauss1DsModel.h.

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

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

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

Definition at line 24 of file PoissonNoise1DObjective.h.

8.9.5.12 `double mappel::MCMCAdaptor1D::eta_bg =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.9.5.13 `double mappel::MCMCAdaptor1D::eta_I =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_I` in the random walk MCMC sampling

Definition at line 31 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()`.

8.9.5.14 `double mappel::MCMCAdaptor1Ds::eta_sigma =-1 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 27 of file MCMCAdaptor1Ds.h.

Referenced by `mappel::MCMCAdaptor1Ds::get_stats()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor1Ds::operator=()`, and `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`.

8.9.5.15 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.9.5.16 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

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

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.9.5.17 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.9.5.18 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

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

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

8.9.5.19 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

8.9.5.20 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

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

8.9.5.21 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static]`,
`[inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

Definition at line 35 of file Gauss1DsMAP.h.

8.9.5.24 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static]`, `[inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.9.5.27 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

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

Definition at line 145 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.9.5.29 `const StringVecT mappel::Gauss1DsModel::prior_types` `[static]`, `[inherited]`

Initial value:

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

Definition at line 50 of file Gauss1DsModel.h.

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

8.9.5.30 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`, `[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.9.5.31 `ImageSizeT` `mappel::ImageFormat1DBase::size` `[protected]`, `[inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

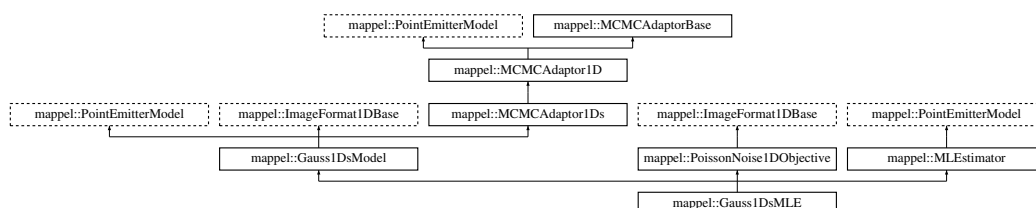
- [Gauss1DsMAP.h](#)
- [Gauss1DsMAP.cpp](#)

8.10 `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)
- [ParamT sample_prior](#) ()
- 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)`
- `void set_intensity_mcmc_sampling (double eta_l=-1)`
- `void set_background_mcmc_sampling (double eta_bg=-1)`
- `void set_mcmc_sigma_scale (double scale)`
- `double get_mcmc_sigma_scale () const`
- `IdxT get_mcmc_num_phases () const`

Static Public Member Functions

- static CompositeDist `make_default_prior (IdxT size, double min_sigma, double max_sigma, const std::string &prior_type)`
- static CompositeDist `make_default_prior_beta_position (IdxT size, double min_sigma, double max_sigma)`
- static CompositeDist `make_default_prior_normal_position (IdxT size, double min_sigma, double max_sigma)`
- static CompositeDist `make_prior_beta_position (IdxT size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma)`
- static CompositeDist `make_prior_normal_position (IdxT size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma)`
- static prior_hessian::TruncatedNormalDist `make_prior_component_position_normal (IdxT size, double pos_↵, sigma=default_sigma_pos)`
- static prior_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta (IdxT size, double pos_↵, beta=default_beta_pos)`
- static prior_hessian::TruncatedGammaDist `make_prior_component_intensity (double mean=default_mean_↵, l, double kappa=default_intensity_kappa)`
- static prior_hessian::TruncatedParetoDist `make_prior_component_sigma (double min_sigma, double max_↵, sigma, double alpha=default_alpha_sigma)`
- static void `set_rng_seed (RngSeedT seed)`
- static `ParallelRngManagerT & get_rng_manager ()`
- static `ParallelRngGeneratorT & get_rng_generator ()`
- static void `check_size (const ImageSizeT &size_)`
Check the size argument for the model.

Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_I](#) = 300
- static const double [default_max_I](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) = 1
- static const [ImageCoordT](#) [global_min_size](#) = 3
- static const [ImageCoordT](#) [global_max_size](#) = 512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [set_mcmc_num_phases](#) ([IdxT](#) [num_phases](#))

Protected Attributes

- CompositeDist [prior](#)
- [IdxT](#) [num_params](#)
- [IdxT](#) [num_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta_sigma](#) = -1
- double [eta_x](#) = 0
- double [eta_I](#) = 0
- double [eta_bg](#) = 0
- [IdxT](#) [num_phases](#)
- double [sigma_scale](#)

8.10.1 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.

8.10.2 Member Typedef Documentation

8.10.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

8.10.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

8.10.2.3 `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.

8.10.2.4 `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.

8.10.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.10.2.6 `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.

8.10.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>` [inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.10.2.8 `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.

8.10.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.10.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.10.2.11 `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.

8.10.2.12 `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.

8.10.2.13 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.10.2.14 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.10.2.15 `using mappel::Gauss1DsModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 47 of file Gauss1DsModel.h.

8.10.3 Constructor & Destructor Documentation

8.10.3.1 `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.

8.10.3.2 `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.

8.10.3.3 `mappel::Gauss1DsMLE::Gauss1DsMLE (ImageSizeT size, CompositeDist && prior)`

Definition at line 20 of file Gauss1DsMLE.cpp.

8.10.3.4 `mappel::Gauss1DsMLE::Gauss1DsMLE (ImageSizeT size, const CompositeDist & prior)`

Definition at line 28 of file Gauss1DsMLE.cpp.

8.10.3.5 `mappel::Gauss1DsMLE::Gauss1DsMLE (const Gauss1DsMLE & o)`

Definition at line 36 of file Gauss1DsMLE.cpp.

8.10.3.6 `mappel::Gauss1DsMLE::Gauss1DsMLE (Gauss1DsMLE && o)`

Definition at line 44 of file Gauss1DsMLE.cpp.

8.10.4 Member Function Documentation

8.10.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const`
[*inherited*]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.10.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [*inherited*]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.10.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.10.4.4 `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`.

8.10.4.5 `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`.

8.10.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.10.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.10.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.10.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.10.4.10 `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()`.

8.10.4.11 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` [inline],[inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.10.4.13 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.10.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` [inline],[inherited]

Definition at line 224 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.10.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` [inline],[inherited]

Definition at line 108 of file ImageFormat1DBase.h.

8.10.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.10.4.17 `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().

8.10.4.18 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.10.4.19 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.10.4.20 `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().

8.10.4.21 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`,`[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.10.4.22 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`,`[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.10.4.23 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().

8.10.4.24 StringVecT mappel::PointEmitterModel::get_param_names () const [inline], [inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.10.4.29 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().

8.10.4.30 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT *idx*) const [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

8.10.4.31 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & *stack*) const [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

8.10.4.32 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().

8.10.4.33 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.10.4.34 bool mappel::PointEmitterModel::has_hyperparam (const std::string & *name*) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.10.4.35 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().

8.10.4.36 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.

8.10.4.37 `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()`.

8.10.4.38 `CompositeDist mappel::Gauss1DsModel::make_default_prior_beta_position (IdxT size, double min_sigma, double max_sigma) [static],[inherited]`

Definition at line 72 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::PointEmitterModel::make_prior_component_sigma()`.

Referenced by `mappel::Gauss1DsModel::make_default_prior()`.

8.10.4.39 `CompositeDist mappel::Gauss1DsModel::make_default_prior_normal_position (IdxT size, double min_sigma, double max_sigma) [static],[inherited]`

Definition at line 83 of file Gauss1DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::PointEmitterModel::make_prior_component_sigma()`.

Referenced by `mappel::Gauss1DsModel::make_default_prior()`.

8.10.4.40 `ImageFormat1DBase::ImageT ImageFormat1DBase::make_image () const [inline],[inherited]`

Definition at line 87 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

8.10.4.41 `ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack (ImageCoordT n) const [inline],[inherited]`

Definition at line 94 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::size`.

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

Definition at line 164 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

8.10.4.43 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const`
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.10.4.44 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.10.4.45 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.10.4.46 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.10.4.47 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.10.4.48 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.10.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.10.4.50 **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().

8.10.4.51 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (*double mean = default_mean_l*, *double kappa = default_intensity_kappa*) [static], [inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::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().

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

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), and mappel::Gauss2DsModel::make_prior_beta_position().

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

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss2DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.10.4.55 `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()`.

8.10.4.56 `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

<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 123 of file Gauss1DsModel.h.

References `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

Referenced by `mappel::Gauss1DsModel::initial_theta_estimate()`.

8.10.4.57 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=().

8.10.4.58 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=().

8.10.4.59 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::I(), and mappel::Gauss1DsModel::Stencil::X.

Referenced by mappel::Gauss1DsModel::pixel_hess_update().

8.10.4.60 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::I(), and mappel::Gauss1DsModel::Stencil::sigma().

8.10.4.61 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::I(), and mappel::Gauss1DsModel::Stencil::sigma().

8.10.4.62 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::I(), mappel::PointEmitterModel::make_param(), mappel::Gauss1DsModel::pixel_grad(), and mappel::Gauss1DsModel::Stencil::sigma().

8.10.4.63 `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`.

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

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.10.4.65 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta)
const [inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.10.4.66 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)
[inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.10.4.67 `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double
step_scale = 1.0) [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`.

8.10.4.68 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)
[inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.10.4.70 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` `[inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.10.4.71 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` `[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

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

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

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.10.4.73 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` `[inline]`, `[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.10.4.74 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` `[inline]`, `[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

8.10.4.75 `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.

8.10.4.76 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1)` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.10.4.77 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.10.4.78 `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()`.

8.10.4.79 `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()`.

8.10.4.80 `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()`.

8.10.4.81 `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`.

8.10.4.82 `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()`.

8.10.4.83 `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()`.

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

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 158 of file PointEmitterModel.cpp.

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

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

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

Definition at line 149 of file PointEmitterModel.cpp.

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

8.10.4.87 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` `[static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.10.4.88 `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()`.

8.10.4.89 `void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & sz)` `[inline],[inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

8.10.4.90 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2Dsxymodel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.10.5 Member Data Documentation

8.10.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static],[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.10.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.10.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.10.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.10.5.5 `const double mappel::PointEmitterModel::default_max_I = INFINITY` [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.10.5.6 `const double mappel::PointEmitterModel::default_mean_I = 300` [static],[inherited]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.10.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` [static],[inherited]

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::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::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.10.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 51 of file Gauss1DsModel.h.

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

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

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

Definition at line 24 of file PoissonNoise1DObjective.h.

8.10.5.12 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.10.5.13 `double mappel::MCMCAdaptor1D::eta_I = 0` `[protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_I` in the random walk MCMC sampling

Definition at line 31 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()`.

8.10.5.14 `double mappel::MCMCAdaptor1Ds::eta_sigma = -1` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 27 of file `MCMCAdaptor1Ds.h`.

Referenced by `mappel::MCMCAdaptor1Ds::get_stats()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor1Ds::operator=()`, and `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`.

8.10.5.15 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.10.5.16 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

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

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.10.5.17 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.10.5.18 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

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

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

8.10.5.19 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

8.10.5.20 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

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

8.10.5.21 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

Definition at line 35 of file `Gauss1DsMLE.h`.

8.10.5.24 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static], [inherited]`

Number of image dimensions.

Definition at line 38 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

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

Definition at line 147 of file `PointEmitterModel.h`.

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

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

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.10.5.27 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

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

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

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

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.10.5.29 `const StringVecT mappel::Gauss1DsModel::prior_types` `[static]`, `[inherited]`

Initial value:

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

Definition at line 50 of file `Gauss1DsModel.h`.

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

8.10.5.30 `double mappel::MCMCAdaptorBase::sigma_scale` [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.10.5.31 `ImageSizeT mappel::ImageFormat1DBase::size` [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 65 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

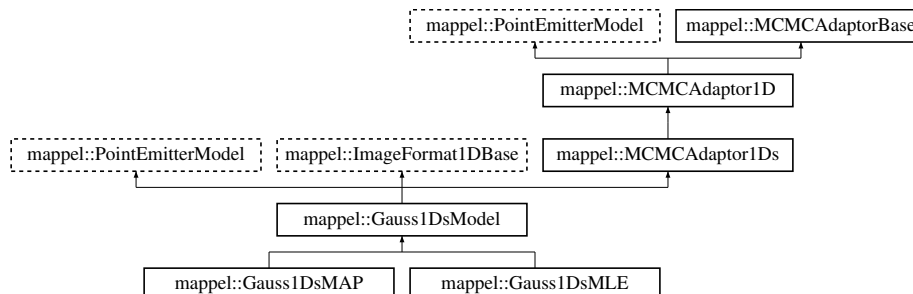
- [Gauss1DsMLE.h](#)
- [Gauss1DsMLE.cpp](#)

8.11 `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)
- `ParamT sample_prior` ()
- 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)
- void [set_intensity_mcmc_sampling](#) (double eta_l=-1)
- void [set_background_mcmc_sampling](#) (double eta_bg=-1)
- void [set_mcmc_sigma_scale](#) (double scale)
- double [get_mcmc_sigma_scale](#) () const
- [IdxT get_mcmc_num_phases](#) () const

Static Public Member Functions

- static CompositeDist [make_default_prior](#) ([IdxT](#) size, double min_sigma, double max_sigma, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) ([IdxT](#) size, double min_sigma, double max_sigma)
- static CompositeDist [make_default_prior_normal_position](#) ([IdxT](#) size, double min_sigma, double max_sigma)
- static CompositeDist [make_prior_beta_position](#) ([IdxT](#) size, double beta_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma)
- static CompositeDist [make_prior_normal_position](#) ([IdxT](#) size, double sigma_xpos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double min_sigma, double max_sigma, double alpha_sigma)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) ([IdxT](#) size, double pos_↵ sigma=default_sigma_pos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↵ beta=default_beta_pos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=default_mean_↵ l, double kappa=default_intensity_kappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵ sigma, double alpha=default_alpha_sigma)
- static void [set_rng_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()
- static void [check_size](#) (const [ImageSizeT](#) &size_)
Check the size argument for the model.

Static Public Attributes

- static const [StringVecT](#) prior_types
- static const std::string DefaultPriorType = "Normal"
- static const std::string DefaultSeperableInitEstimator = "TrustRegion"
- static const double bounds_epsilon = 1.0E-6
- static const double global_min_psf_sigma = 1E-1
- static const double global_max_psf_sigma = 1E2
- static const double default_beta_pos = 3
- static const double default_sigma_pos = 1

- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2
- static const `ImageCoordT` `num_dim` = 1
- static const `ImageCoordT` `global_min_size` = 3
- static const `ImageCoordT` `global_max_size` = 512
- static const double `global_default_mcmc_sigma_scale` = 0.05
- static const double `global_max_mcmc_sigma_scale` = 0.5

Protected Member Functions

- `Gauss1DsModel` (`IdxT` size_)
- `Gauss1DsModel` (const `Gauss1DsModel` &o)
- `Gauss1DsModel` (`Gauss1DsModel` &&o)
- `Gauss1DsModel` & `operator=` (const `Gauss1DsModel` &o)
- `Gauss1DsModel` & `operator=` (`Gauss1DsModel` &&o)
- void `set_mcmc_num_phases` (`IdxT` num_phases)

Protected Attributes

- CompositeDist `prior`
- `IdxT` `num_params`
- `IdxT` `num_hyperparams`
- `ParamT` `lbound`
- `ParamT` `ubound`
- `ImageSizeT` `size`
- double `eta_sigma` == -1
- double `eta_x` = 0
- double `eta_l` = 0
- double `eta_bg` = 0
- `IdxT` `num_phases`
- double `sigma_scale`

8.11.1 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`.

8.11.2 Member Typedef Documentation

8.11.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 25 of file `ImageFormat1DBase.h`.

8.11.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

8.11.2.3 `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.

8.11.2.4 `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.

8.11.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
`[inherited]`

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.11.2.6 `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.

8.11.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.11.2.8 `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.

8.11.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.11.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.11.2.11 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.11.2.12 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.11.2.13 `using mappel::Gauss1DsModel::StencilVecT = std::vector<Stencil>`

Definition at line 47 of file Gauss1DsModel.h.

8.11.3 Constructor & Destructor Documentation

8.11.3.1 `mappel::Gauss1DsModel::Gauss1DsModel (IdxT size_)` `[explicit], [protected]`

Definition at line 12 of file Gauss1DsModel.cpp.

8.11.3.2 `mappel::Gauss1DsModel::Gauss1DsModel (const Gauss1DsModel & o)` `[protected]`

Definition at line 17 of file Gauss1DsModel.cpp.

8.11.3.3 `mappel::Gauss1DsModel::Gauss1DsModel (Gauss1DsModel && o)` `[protected]`

Definition at line 22 of file Gauss1DsModel.cpp.

8.11.4 Member Function Documentation

8.11.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.11.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.11.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.11.4.4 `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`.

8.11.4.5 `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`.

8.11.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.11.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` `[inherited]`

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.11.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` `[inherited]`

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.11.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.11.4.10 `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().

8.11.4.11 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline], [inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.11.4.13 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.11.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.11.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

8.11.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.11.4.17 `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().

8.11.4.18 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.11.4.19 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.11.4.20 `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().

8.11.4.21 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const [inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.11.4.22 `IdxT mappel::PointEmitterModel::get_num_params () const [inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.11.4.23 `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().

8.11.4.24 `StringVecT mappel::PointEmitterModel::get_param_names () const [inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.11.4.29 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().

8.11.4.30 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT idx) const [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

8.11.4.31 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

8.11.4.32 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().

8.11.4.33 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.11.4.34 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const` `[inline]`, `[inherited]`

Definition at line 228 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.11.4.35 `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()`.

8.11.4.36 `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`.

8.11.4.37 `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()`.

8.11.4.38 `CompositeDist mappel::Gauss1DsModel::make_default_prior_beta_position (IdxT size, double min_sigma, double max_sigma)` `[static]`

Definition at line 72 of file `Gauss1DsModel.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::PointEmitterModel::make_prior_component_sigma()`.

Referenced by `make_default_prior()`.

8.11.4.39 `CompositeDist mappel::Gauss1DsModel::make_default_prior_normal_position (IdxT size, double min_sigma, double max_sigma)` `[static]`

Definition at line 83 of file `Gauss1DsModel.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::PointEmitterModel::make_prior_component_sigma()`.

Referenced by `make_default_prior()`.

8.11.4.40 ImageFormat1DBase::ImageT ImageFormat1DBase::make_image () const [inline],[inherited]

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

8.11.4.41 ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack (ImageCoordT n) const
[inline],[inherited]

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

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

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.11.4.43 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const
[inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.11.4.44 MatT mappel::PointEmitterModel::make_param_mat () const [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.11.4.45 template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.11.4.46 CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.11.4.47 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.11.4.48 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline]`,
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.11.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.11.4.50 `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().

8.11.4.51 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.11.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos) [static], [inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

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

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `make_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()`.

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.11.4.55 `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()`.

8.11.4.56 `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()`.

8.11.4.57 `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=()`.

8.11.4.58 `Gauss1DsModel & mappel::Gauss1DsModel::operator= (Gauss1DsModel && o)` `[protected]`

Definition at line 35 of file Gauss1DsModel.cpp.

References `DefaultPriorType`, `mappel::MCMCAdaptor1Ds::operator=()`, and `prior_types`.

8.11.4.59 `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()`.

8.11.4.60 `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()`.

8.11.4.61 `void mappel::Gauss1DsModel::pixel_hess (IdxT i, const Stencil & s, MatT & hess) const [inline]`

Definition at line 160 of file Gauss1DsModel.h.

References `mappel::Gauss1DsModel::Stencil::DX`, `mappel::Gauss1DsModel::Stencil::DXS`, `mappel::Gauss1DsModel::Stencil::DXS2`, `mappel::Gauss1DsModel::Stencil::DXSX`, `mappel::Gauss1DsModel::Stencil::l()`, and `mappel::Gauss1DsModel::Stencil::sigma()`.

8.11.4.62 `void mappel::Gauss1DsModel::pixel_hess_update (IdxT i, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess) const`

Definition at line 211 of file Gauss1DsModel.cpp.

References `mappel::Gauss1DsModel::Stencil::DX`, `mappel::Gauss1DsModel::Stencil::DXS`, `mappel::Gauss1DsModel::Stencil::DXS2`, `mappel::Gauss1DsModel::Stencil::DXSX`, `mappel::Gauss1DsModel::Stencil::l()`, `mappel::PointEmitterModel::make_param()`, `pixel_grad()`, and `mappel::Gauss1DsModel::Stencil::sigma()`.

8.11.4.63 `double mappel::Gauss1DsModel::pixel_model_value (IdxT i, const Stencil & s) const [inline]`

Definition at line 134 of file Gauss1DsModel.h.

References `mappel::Gauss1DsModel::Stencil::bg()`, `mappel::Gauss1DsModel::Stencil::l()`, and `mappel::Gauss1DsModel::Stencil::X`.

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

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.11.4.65 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const [inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.11.4.66 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name) [inline], [inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.11.4.67 `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)` [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`.

8.11.4.68 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)` [inherited]

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.11.4.70 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.11.4.71 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.11.4.73 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.11.4.74 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams) [inline], [inherited]`

Definition at line 220 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::set_hyperparams(), and mappel::Gauss2DsModel::set_hyperparams().

8.11.4.75 `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.

8.11.4.76 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_mean_I, mappel::MCMCAdaptor1D::eta_I, mappel::PointEmitterModel::get_hyperparam_value(), and mappel::MCMCAdaptorBase::sigma_scale.

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

8.11.4.77 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by set_min_sigma().

8.11.4.78 `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().

8.11.4.79 void mappel::Gauss1DsModel::set_max_sigma (const VecT & *max_sigma*)

Definition at line 153 of file Gauss1DsModel.cpp.

References set_max_sigma().

8.11.4.80 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().

8.11.4.81 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.

8.11.4.82 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().

8.11.4.83 void mappel::Gauss1DsModel::set_min_sigma (const VecT & *min_sigma*)

Definition at line 148 of file Gauss1DsModel.cpp.

References set_min_sigma().

8.11.4.84 void mappel::PointEmitterModel::set_param_names (const StringVecT & *desc*) [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.11.4.85 void mappel::PointEmitterModel::set_prior (CompositeDist && *prior*) [inherited]

Definition at line 158 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_hyperparams, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::set_prior(), and mappel::Gauss2DsModel::set_prior().

8.11.4.86 void mappel::PointEmitterModel::set_prior (const CompositeDist & *prior_*) [inherited]

Definition at line 149 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_hyperparams, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

8.11.4.87 void mappel::PointEmitterModel::set_rng_seed (RngSeedT *seed*) [static],[inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.11.4.88 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().

8.11.4.89 void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & *sz*) [inline],[inherited]

Definition at line 75 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::set_size().

8.11.4.90 void mappel::PointEmitterModel::set_ubound (const ParamT & *ubound*) [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by set_max_sigma(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.11.4.91 bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & *theta*) const [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::make_stencil(), make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DsModel::make_stencil(), mappel::Gauss2DsModel::make_stencil(), and mappel::PointEmitterModel::theta_stack_in_bounds().

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

Definition at line 297 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

8.11.5 Member Data Documentation

8.11.5.1 const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static],[inherited]

Distance from the boundary to constrain in bound_theta and bounded_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss2↔DsModel::set_max_sigma_ratio(), and mappel::PointEmitterModel::set_ubound().

8.11.5.2 const double mappel::PointEmitterModel::default_alpha_sigma = 2 [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.11.5.3 const double mappel::PointEmitterModel::default_beta_pos = 3 [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.11.5.4 const double mappel::PointEmitterModel::default_intensity_kappa = 2 [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.11.5.5 const double mappel::PointEmitterModel::default_max_l = INFINITY [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.11.5.6 const double mappel::PointEmitterModel::default_mean_l = 300 [static],[inherited]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.11.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.11.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 51 of file Gauss1DsModel.h.

Referenced by `operator=()`.

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

8.11.5.11 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.11.5.12 `double mappel::MCMCAdaptor1D::eta_I = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_I` in the random walk MCMC sampling

Definition at line 31 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()`.

8.11.5.13 `double mappel::MCMCAdaptor1Ds::eta_sigma = 1` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 27 of file `MCMCAdaptor1Ds.h`.

Referenced by `mappel::MCMCAdaptor1Ds::get_stats()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor1Ds::operator=()`, and `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`.

8.11.5.14 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.11.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

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

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.11.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.11.5.17 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

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

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

8.11.5.18 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

8.11.5.19 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static]`, `[inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

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

8.11.5.20 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static]`, `[inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.11.5.22 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static]`, `[inherited]`

Number of image dimensions.

Definition at line 38 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

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

Definition at line 147 of file `PointEmitterModel.h`.

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

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

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.11.5.25 `IdxT mappel::MCMCAdaptorBase::num_phases` [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

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

Definition at line 145 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `get_max_sigma()`, `get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `set_max_sigma()`, `set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

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

Initial value:

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

Definition at line 50 of file Gauss1DsModel.h.

Referenced by `operator=()`.

8.11.5.28 `double mappel::MCMCAdaptorBase::sigma_scale` [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.11.5.29 `ImageSizeT` `mappel::ImageFormat1DBase::size` `[protected]`, `[inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

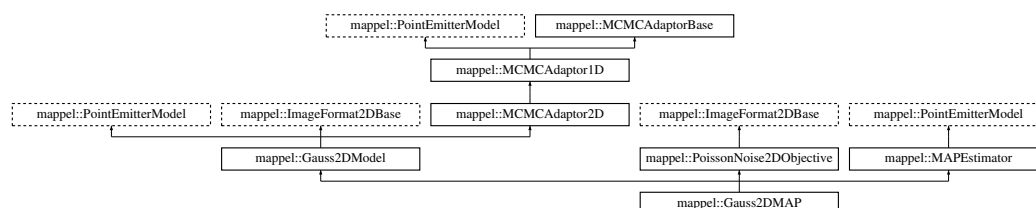
- [Gauss1DsModel.h](#)
- [Gauss1DsModel.cpp](#)

8.12 mappel::Gauss2DMP Class Reference

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

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DMP.h>
```

Inheritance diagram for `mappel::Gauss2DMP`:



Public Types

- using [Gauss1DSumModelT](#) = [Gauss1DMAP](#)
- using [StencilVecT](#) = `std::vector< Stencil >`
- using [ParamT](#) = `arma::vec`
- using [ParamVecT](#) = `arma::mat`
- using [ImageCoordT](#) = `uint32_t`
- using [ImagePixelT](#) = `double`
- `template<class CoordT >`
using [ImageSizeShapeT](#) = `arma::Col< CoordT >`
- `template<class CoordT >`
using [ImageSizeVecShapeT](#) = `arma::Mat< CoordT >`
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- `template<class PixelT >`
using [ImageShapeT](#) = `arma::Mat< PixelT >`
- `template<class PixelT >`
using [ImageStackShapeT](#) = `arma::Cube< PixelT >`
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >
- using [ModelDataT](#) = [ImageT](#)
- using [ModelDataStackT](#) = [ImageStackT](#)

Public Member Functions

- [Gauss2DMAP](#) ([ImageCoordT](#) size, double [psf_sigma](#), const `std::string` &prior_type=[DefaultPriorType](#))
- [Gauss2DMAP](#) (const [ImageSizeT](#) &size, double [psf_sigma](#), const `std::string` &prior_type=[DefaultPriorType](#))
- `template<class IntType , class FloatType >`
[Gauss2DMAP](#) (const `arma::Col< IntType >` &size, const `arma::Col< FloatType >` &[psf_sigma](#), const `std::string` &prior_type=[DefaultPriorType](#))
- [Gauss2DMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &[psf_sigma](#), [CompositeDist](#) &&prior)
- [Gauss2DMAP](#) ([ImageSizeT](#) &&size, [VecT](#) &&[psf_sigma](#), [CompositeDist](#) &&prior)
- [Gauss2DMAP](#) (const [ImageSizeT](#) &size, const [VecT](#) &[psf_sigma](#), const [CompositeDist](#) &prior)
- [Gauss2DMAP](#) (const [Gauss2DMAP](#) &o)
- [Gauss2DMAP](#) & operator= (const [Gauss2DMAP](#) &o)
- [Gauss2DMAP](#) ([Gauss2DMAP](#) &&o)
- [Gauss2DMAP](#) & operator= ([Gauss2DMAP](#) &&o)
- void [set_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set_prior](#) ([CompositeDist](#) &&prior_)
- void [set_prior](#) (const [CompositeDist](#) &prior_)
- void [set_size](#) (const [ImageSizeT](#) &size_)
- const [VecT](#) & [get_psf_sigma](#) () const
- double [get_psf_sigma](#) ([IdxT](#) idx) const
- void [set_psf_sigma](#) (double new_psf_sigma)
- void [set_psf_sigma](#) (const [VecT](#) &new_psf_sigma)
- [StatsT](#) [get_stats](#) () const
- [Stencil](#) [make_stencil](#) (const [ParamT](#) &theta, bool compute_derivatives=true) const
Make a new Model::Stencil object at theta.
- double [pixel_model_value](#) (int i, int j, const [Stencil](#) &s) const
- void [pixel_grad](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad) const

- void `pixel_grad2` (int i, int j, const `Stencil` &s, `ParamT` &pgrad2) const
- void `pixel_hess` (int i, int j, const `Stencil` &s, `MatT` &hess) const
- void `pixel_hess_update` (int i, int j, const `Stencil` &s, double dm_ratio_m1, double dmm_ratio, `ParamT` &grad, `MatT` &hess) const
- pixel derivative inner loop calculations.*
- `Stencil initial_theta_estimate` (const `ImageT` &im)
- 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)
- `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)
- `ParamT sample_prior` ()
- 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)
- void [set_intensity_mcmc_sampling](#) (double eta_l=-1)
- void [set_background_mcmc_sampling](#) (double eta_bg=-1)
- void [set_mcmc_sigma_scale](#) (double scale)
- double [get_mcmc_sigma_scale](#) () const
- [IdxT](#) [get_mcmc_num_phases](#) () const

Static Public Member Functions

- static CompositeDist [make_default_prior](#) (const [ImageSizeT](#) &size, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) (const [ImageSizeT](#) &size)
- static CompositeDist [make_default_prior_normal_position](#) (const [ImageSizeT](#) &size)
- static CompositeDist [make_prior_beta_position](#) (const [ImageSizeT](#) &size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
- static CompositeDist [make_prior_normal_position](#) (const [ImageSizeT](#) &size, double sigma_xpos, double beta_↵_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) ([IdxT](#) size, double pos_↵_sigma=[default_sigma_pos](#))
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↵_beta=[default_beta_pos](#))
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[default_mean_↵_l](#), double kappa=[default_intensity_kappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵_sigma, double alpha=[default_alpha_sigma](#))
- static void [set_rng_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()
- static void [check_size](#) (const [ImageSizeT](#) &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) =2
- static const [ImageCoordT](#) [global_min_size](#) =3
- static const [ImageCoordT](#) [global_max_size](#) =512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [update_internal_1Dsum_estimators](#) ()
- void [set_mcmc_num_phases](#) ([IdxT](#) num_phases)

Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make_internal_1Dsum_estimator](#) ([IdxT](#) dim, const [ImageSizeT](#) &size, const [VecT](#) &psf_sigma, const [CompositeDist](#) &prior)

Protected Attributes

- [VecT](#) [psf_sigma](#)
- [Gauss1DSumModelT](#) [x_model](#)
- [Gauss1DSumModelT](#) [y_model](#)
- [CompositeDist](#) [prior](#)
- [IdxT](#) [num_params](#)
- [IdxT](#) [num_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta_y](#) =0
- double [eta_x](#) =0
- double [eta_l](#) =0
- double [eta_bg](#) =0
- [IdxT](#) [num_phases](#)
- double [sigma_scale](#)

8.12.1 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.

8.12.2 Member Typedef Documentation

8.12.2.1 `using mappel::Gauss2DModel::Gauss1DSumModelT = Gauss1DMAP` [inherited]

Definition at line 23 of file Gauss2DModel.h.

8.12.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

8.12.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

8.12.2.4 `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.

8.12.2.5 `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.

8.12.2.6 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
[inherited]

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

8.12.2.7 `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.

8.12.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.12.2.9 `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.

8.12.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.12.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.12.2.12 `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.

8.12.2.13 `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.

8.12.2.14 using mappel::PointEmitterModel::ParamT = arma::vec [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.12.2.15 using mappel::PointEmitterModel::ParamVecT = arma::mat [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.12.2.16 using mappel::Gauss2DModel::StencilVecT = std::vector<Stencil> [inherited]

Definition at line 47 of file Gauss2DModel.h.

8.12.3 Constructor & Destructor Documentation

8.12.3.1 mappel::Gauss2DMap::Gauss2DMap (ImageCoordT size, double psf_sigma, const std::string & prior_type = DefaultPriorType)

Definition at line 11 of file Gauss2DMap.cpp.

8.12.3.2 mappel::Gauss2DMap::Gauss2DMap (const ImageSizeT & size, double psf_sigma, const std::string & prior_type = DefaultPriorType)

Definition at line 15 of file Gauss2DMap.cpp.

8.12.3.3 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.

8.12.3.4 mappel::Gauss2DMap::Gauss2DMap (const ImageSizeT & size, const VecT & psf_sigma, CompositeDist && prior)

Definition at line 27 of file Gauss2DMap.cpp.

8.12.3.5 mappel::Gauss2DMap::Gauss2DMap (ImageSizeT && size, VecT && psf_sigma, CompositeDist && prior)

Definition at line 19 of file Gauss2DMap.cpp.

8.12.3.6 mappel::Gauss2DMap::Gauss2DMap (const ImageSizeT & size, const VecT & psf_sigma, const CompositeDist & prior)

Definition at line 35 of file Gauss2DMap.cpp.

8.12.3.7 mappel::Gauss2DMap::Gauss2DMap (const Gauss2DMap & o)

Definition at line 43 of file Gauss2DMap.cpp.

8.12.3.8 mappel::Gauss2DMap::Gauss2DMap (Gauss2DMap && o)

Definition at line 51 of file Gauss2DMap.cpp.

8.12.4 Member Function Documentation

8.12.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

8.12.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

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

8.12.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::make_param_stack().

8.12.4.4 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.

8.12.4.5 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.

8.12.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.12.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.12.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.12.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.12.4.10 `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().

8.12.4.11 `Gauss1DSumModelT mappel::Gauss2DModel::debug_internal_sum_model_x () const` [inline], [inherited]

Definition at line 89 of file Gauss2DModel.h.

References mappel::Gauss2DModel::x_model.

8.12.4.12 `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`.

8.12.4.13 `int` `mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

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

References `mappel::PointEmitterModel::prior`.

8.12.4.15 `double` `mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.12.4.16 `PointEmitterModel::ParamT` `mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.12.4.17 `ImageFormat2DBase::ImageT` `mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 106 of file `ImageFormat2DBase.h`.

8.12.4.18 `const PointEmitterModel::ParamT &` `mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.12.4.19 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

8.12.4.20 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

8.12.4.21 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

8.12.4.22 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.12.4.23 `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()`.

8.12.4.24 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 200 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.12.4.27 `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().

8.12.4.28 `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.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.12.4.31 `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const` `[inline],[inherited]`

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.12.4.32 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT idx) const` `[inherited]`

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

8.12.4.33 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

8.12.4.34 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().

8.12.4.35 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.12.4.36 bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.12.4.37 Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate (const ImageT & im) [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().

8.12.4.38 Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init) [inline],[inherited]

Definition at line 201 of file Gauss2DModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::Gauss2DModel::initial_theta_estimate().

8.12.4.39 `Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init, const std::string & estimator)` `[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::PointEmitterModel::theta_in_bounds()`, `mappel::PointEmitterModel::ubound`, `mappel::Gauss2DModel::x_model`, and `mappel::Gauss2DModel::y_model`.

8.12.4.40 `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()`.

8.12.4.41 `CompositeDist mappel::Gauss2DModel::make_default_prior_beta_position (const ImageSizeT & size)` `[static]`, `[inherited]`

Definition at line 171 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::ImageFormat2DBase::size`.

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

8.12.4.42 `CompositeDist mappel::Gauss2DModel::make_default_prior_normal_position (const ImageSizeT & size)` `[static]`, `[inherited]`

Definition at line 182 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::ImageFormat2DBase::size`.

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

8.12.4.43 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const` `[inline]`, `[inherited]`

Definition at line 85 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.12.4.44 `ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 92 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.12.4.45 `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()`.

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

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

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

8.12.4.47 `template<class FillT > PointEmitterModel::ParamT` `mappel::PointEmitterModel::make_param (FillT fill) const` `[inherited]`

Definition at line 181 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.12.4.48 `MatT` `mappel::PointEmitterModel::make_param_mat () const` `[inline], [inherited]`

Definition at line 172 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.12.4.49 `template<class FillT > MatT` `mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.12.4.50 `CubeT` `mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline], [inherited]`

Definition at line 176 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.12.4.51 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.12.4.52 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline]`,
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.12.4.53 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.12.4.54 `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.

8.12.4.55 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.12.4.56 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos) [static],[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

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

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

Definition at line 85 of file PointEmitterModel.cpp.

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

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.12.4.59 `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`.

8.12.4.60 `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>	Prameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

Returns

A new [Stencil](#) object ready to compute with

Definition at line 131 of file Gauss2DModel.h.

References [mappel::Gauss2DModel::Stencil::Stencil\(\)](#), and [mappel::PointEmitterModel::theta_in_bounds\(\)](#).

Referenced by [mappel::Gauss2DModel::initial_theta_estimate\(\)](#).

8.12.4.61 Gauss2DMAP & mappel::Gauss2DMAP::operator= (const Gauss2DMAP & o)

Definition at line 59 of file Gauss2DMAP.cpp.

References [mappel::MAPEstimator::operator=\(\)](#), [mappel::PoissonNoise2DObjective::operator=\(\)](#), [mappel::Image↔Format2DBase::operator=\(\)](#), [mappel::Gauss2DModel::operator=\(\)](#), and [mappel::PointEmitterModel::operator=\(\)](#).

8.12.4.62 Gauss2DMAP & mappel::Gauss2DMAP::operator= (Gauss2DMAP && o)

Definition at line 70 of file Gauss2DMAP.cpp.

References [mappel::MAPEstimator::operator=\(\)](#), [mappel::PoissonNoise2DObjective::operator=\(\)](#), [mappel::Image↔Format2DBase::operator=\(\)](#), [mappel::Gauss2DModel::operator=\(\)](#), and [mappel::PointEmitterModel::operator=\(\)](#).

8.12.4.63 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\(\)](#).

8.12.4.64 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](#).

8.12.4.65 `void mappel::Gauss2DModel::pixel_hess (int i, int j, const Stencil & s, MatT & hess) const` `[inline]`,
`[inherited]`

Definition at line 181 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

8.12.4.66 `void mappel::Gauss2DModel::pixel_hess_update (int i, int j, const Stencil & s, double dm_ratio_m1, double dmm_ratio, ParamT & grad, MatT & hess) const` `[inherited]`

pixel derivative inner loop calculations.

Definition at line 282 of file Gauss2DModel.cpp.

References `mappel::Gauss2DModel::Stencil::DX`, `mappel::Gauss2DModel::Stencil::DXS`, `mappel::Gauss2DModel::Stencil::DY`, `mappel::Gauss2DModel::Stencil::DYS`, `mappel::Gauss2DModel::Stencil::l()`, `mappel::PointEmitterModel::make_param()`, `mappel::Gauss2DModel::pixel_grad()`, `mappel::Gauss2DModel::psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

8.12.4.67 `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`.

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

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.12.4.69 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const` `[inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.12.4.70 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)` `[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.12.4.71 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0) [inherited]`

Definition at line 59 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_u`, `x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

8.12.4.72 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng) [inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.12.4.74 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1) [inherited]`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.12.4.75 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_) [inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.12.4.77 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.12.4.78 `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().

8.12.4.79 `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.

8.12.4.80 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_mean_I, mappel::MCMCAdaptor1D::eta_I, mappel::PointEmitter↵ Model::get_hyperparam_value(), and mappel::MCMCAdaptorBase::sigma_scale.

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

8.12.4.81 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss1DsModel::set_min_sigma().

8.12.4.82 `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().

8.12.4.83 `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.

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

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.12.4.85 `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()`.

8.12.4.86 `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()`.

8.12.4.87 `void mappel::Gauss2DModel::set_psf_sigma (double new_psf_sigma) [inline],[inherited]`

Definition at line 146 of file Gauss2DModel.h.

8.12.4.88 `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`.

8.12.4.89 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.12.4.90 `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`.

8.12.4.91 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::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()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2Dsxymodel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.12.4.94 `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()`.

8.12.5 Member Data Documentation

8.12.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static]`, `[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.12.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static]`, `[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.12.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.12.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.12.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.12.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.12.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.12.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 51 of file Gauss2DModel.h.

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

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

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

Definition at line 23 of file PoissonNoise2DObjective.h.

8.12.5.12 `double mappel::MCMCAdaptor1D::eta_bg =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.12.5.13 `double mappel::MCMCAdaptor1D::eta_l =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.12.5.14 `double mappel::MCMCAdaptor1D::eta_x =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.12.5.15 `double mappel::MCMCAdaptor2D::eta_y = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 27 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()`.

8.12.5.16 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

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

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.12.5.17 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.12.5.18 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

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

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

8.12.5.19 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file `ImageFormat2DBase.h`.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

8.12.5.20 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

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

8.12.5.21 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size =3` `[static]`,
`[inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

Definition at line 37 of file Gauss2DMap.h.

8.12.5.24 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim =2` `[static]`,
`[inherited]`

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.12.5.27 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

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

Definition at line 145 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.12.5.29 `const StringVecT mappel::Gauss2DModel::prior_types` `[static]`, `[inherited]`

Initial value:

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

Definition at line 50 of file Gauss2DModel.h.

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

8.12.5.30 `VecT mappel::Gauss2DModel::psf_sigma` `[protected]`, `[inherited]`

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 104 of file Gauss2DModel.h.

Referenced by `mappel::cgauss_compute_estimate()`, `mappel::cgauss_compute_estimate_debug()`, `mappel::cgauss_heuristic_compute_estimate()`, `mappel::Gauss2DModel::Stencil::compute_derivatives()`, `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DModel::get_psf_sigma()`, `mappel::Gauss2DModel::make_internal_1Dsum_estimator()`, `mappel::Gauss2DModel::operator=()`, `mappel::Gauss2DModel::pixel_grad2()`, `mappel::Gauss2DModel::pixel_hess()`, `mappel::Gauss2DModel::pixel_hess_update()`, `mappel::Gauss2DModel::set_psf_sigma()`, `mappel::Gauss2DModel::Stencil::Stencil()`, and `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`.

8.12.5.31 double mappel::MCMCAdaptorBase::sigma_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_sigma_scale(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling(), and mappel::MCMCAdaptorBase::set_mcmc_sigma_scale().

8.12.5.32 ImageSizeT mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss_compute_estimate(), mappel::cgauss_compute_estimate_debug(), mappel::cgauss_heuristic_compute_estimate(), mappel::ImageFormat2DBase::check_image_shape(), mappel::Gauss2DModel::Stencil::compute_derivatives(), mappel::Gauss2DsModel::Stencil::compute_derivatives(), mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::methods::expected_information(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsMAP::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().

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.12.5.34 Gauss1DSumModelT mappel::Gauss2DModel::x_model [protected],[inherited]

X-model fits 2D images X-axis (column sum)

Definition at line 105 of file Gauss2DModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_x(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::set_psf_sigma(), mappel::Gauss2DModel::set_size(), and mappel::Gauss2DModel::update_internal_1Dsum_estimators().

8.12.5.35 Gauss1DSumModelT mappel::Gauss2DModel::y_model [protected],[inherited]

Y-model fits 2D images Y-axis (row sum)

Definition at line 106 of file Gauss2DModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::set_psf_sigma(), mappel::Gauss2DModel::set_size(), and mappel::Gauss2DModel::update_internal_1Dsum_estimators().

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

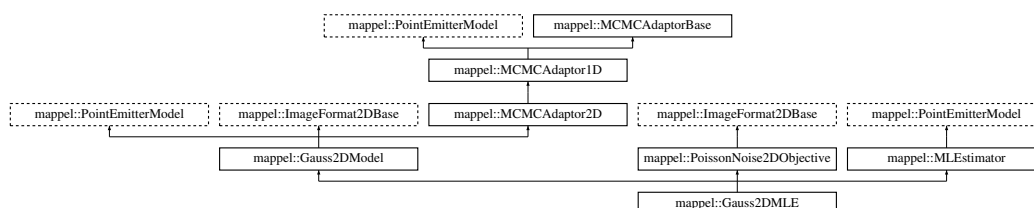
- [Gauss2DMAP.h](#)
- [Gauss2DMAP.cpp](#)

8.13 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)
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)
- `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)
- `ParamT sample_prior` ()
- 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)
- void [set_intensity_mcmc_sampling](#) (double eta_l=-1)
- void [set_background_mcmc_sampling](#) (double eta_bg=-1)
- void [set_mcmc_sigma_scale](#) (double scale)
- double [get_mcmc_sigma_scale](#) () const
- [IdxT](#) [get_mcmc_num_phases](#) () const

Static Public Member Functions

- static CompositeDist [make_default_prior](#) (const [ImageSizeT](#) &size, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) (const [ImageSizeT](#) &size)
- static CompositeDist [make_default_prior_normal_position](#) (const [ImageSizeT](#) &size)
- static CompositeDist [make_prior_beta_position](#) (const [ImageSizeT](#) &size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
- static CompositeDist [make_prior_normal_position](#) (const [ImageSizeT](#) &size, double sigma_xpos, double beta_↵_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) ([IdxT](#) size, double pos_↵_sigma=[default_sigma_pos](#))
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↵_beta=[default_beta_pos](#))
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[default_mean_↵_l](#), double kappa=[default_intensity_kappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵_sigma, double alpha=[default_alpha_sigma](#))
- static void [set_rng_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()
- static void [check_size](#) (const [ImageSizeT](#) &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) =2
- static const [ImageCoordT](#) [global_min_size](#) =3
- static const [ImageCoordT](#) [global_max_size](#) =512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [update_internal_1Dsum_estimators](#) ()
- void [set_mcmc_num_phases](#) ([IdxT](#) num_phases)

Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make_internal_1Dsum_estimator](#) ([IdxT](#) dim, const [ImageSizeT](#) &size, const [VecT](#) &psf_sigma, const [CompositeDist](#) &prior)

Protected Attributes

- [VecT](#) [psf_sigma](#)
- [Gauss1DSumModelT](#) [x_model](#)
- [Gauss1DSumModelT](#) [y_model](#)
- [CompositeDist](#) [prior](#)
- [IdxT](#) [num_params](#)
- [IdxT](#) [num_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta_y](#) =0
- double [eta_x](#) =0
- double [eta_l](#) =0
- double [eta_bg](#) =0
- [IdxT](#) [num_phases](#)
- double [sigma_scale](#)

8.13.1 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.

8.13.2 Member Typedef Documentation

8.13.2.1 `using mappel::Gauss2DModel::Gauss1DSumModelT = Gauss1DMap` `[inherited]`

Definition at line 23 of file Gauss2DModel.h.

8.13.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` `[inherited]`

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

8.13.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

8.13.2.4 `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.

8.13.2.5 `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.

8.13.2.6 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
`[inherited]`

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

8.13.2.7 `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.

8.13.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.13.2.9 `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.

8.13.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.13.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.13.2.12 `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.

8.13.2.13 `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.

8.13.2.14 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.13.2.15 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.13.2.16 `using mappel::Gauss2DModel::StencilVecT = std::vector<Stencil>` [inherited]

Definition at line 47 of file Gauss2DModel.h.

8.13.3 Constructor & Destructor Documentation

8.13.3.1 `mappel::Gauss2DMLE::Gauss2DMLE (ImageCoordT size, double psf_sigma, const std::string & prior_type = DefaultPriorType)`

Definition at line 11 of file Gauss2DMLE.cpp.

8.13.3.2 `mappel::Gauss2DMLE::Gauss2DMLE (const ImageSizeT & size, double psf_sigma, const std::string & prior_type = DefaultPriorType)`

Definition at line 15 of file Gauss2DMLE.cpp.

8.13.3.3 `mappel::Gauss2DMLE::Gauss2DMLE (const ImageSizeT & size, const VecT & psf_sigma, const std::string & prior_type = DefaultPriorType)`

Definition at line 19 of file Gauss2DMLE.cpp.

8.13.3.4 `mappel::Gauss2DMLE::Gauss2DMLE (const ImageSizeT & size, const VecT & psf_sigma, CompositeDist && prior)`

Definition at line 23 of file Gauss2DMLE.cpp.

8.13.3.5 `mappel::Gauss2DMLE::Gauss2DMLE (const ImageSizeT & size, const VecT & psf_sigma, const CompositeDist & prior)`

Definition at line 31 of file Gauss2DMLE.cpp.

8.13.3.6 `mappel::Gauss2DMLE::Gauss2DMLE (const Gauss2DMLE & o)`

Definition at line 39 of file Gauss2DMLE.cpp.

8.13.3.7 mappel::Gauss2DMLE::Gauss2DMLE (Gauss2DMLE && o)

Definition at line 47 of file Gauss2DMLE.cpp.

8.13.4 Member Function Documentation

8.13.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

8.13.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

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

8.13.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::make_param_stack().

8.13.4.4 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.

8.13.4.5 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.

8.13.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.13.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.13.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.13.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.13.4.10 `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().

8.13.4.11 `Gauss1DSumModelT mappel::Gauss2DModel::debug_internal_sum_model_x () const` [inline], [inherited]

Definition at line 89 of file Gauss2DModel.h.

References mappel::Gauss2DModel::x_model.

8.13.4.12 `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`.

8.13.4.13 `int` `mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

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

References `mappel::PointEmitterModel::prior`.

8.13.4.15 `double` `mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.13.4.16 `PointEmitterModel::ParamT` `mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.13.4.17 `ImageFormat2DBase::ImageT` `mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 106 of file `ImageFormat2DBase.h`.

8.13.4.18 `const PointEmitterModel::ParamT &` `mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.13.4.19 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

8.13.4.20 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

8.13.4.21 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

8.13.4.22 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.13.4.23 `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()`.

8.13.4.24 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 200 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

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

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.13.4.27 `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()`.

8.13.4.28 `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`.

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

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

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

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.13.4.31 `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const` `[inline],[inherited]`

Definition at line 74 of file `ImageFormat2DBase.h`.

References `mappel::ImageFormat2DBase::size`.

8.13.4.32 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT idx) const` `[inherited]`

Definition at line 41 of file `ImageFormat2DBase.cpp`.

References `mappel::ImageFormat2DBase::size`.

8.13.4.33 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

8.13.4.34 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().

8.13.4.35 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.13.4.36 bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.13.4.37 Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate (const ImageT & im) [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().

8.13.4.38 Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init) [inline],[inherited]

Definition at line 201 of file Gauss2DModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::Gauss2DModel::initial_theta_estimate().

8.13.4.39 `Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init, const std::string & estimator)` `[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::PointEmitterModel::theta_in_bounds()`, `mappel::PointEmitterModel::ubound`, `mappel::Gauss2DModel::x_model`, and `mappel::Gauss2DModel::y_model`.

8.13.4.40 `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()`.

8.13.4.41 `CompositeDist mappel::Gauss2DModel::make_default_prior_beta_position (const ImageSizeT & size)` `[static]`, `[inherited]`

Definition at line 171 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::ImageFormat2DBase::size`.

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

8.13.4.42 `CompositeDist mappel::Gauss2DModel::make_default_prior_normal_position (const ImageSizeT & size)` `[static]`, `[inherited]`

Definition at line 182 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::ImageFormat2DBase::size`.

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

8.13.4.43 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const` `[inline]`, `[inherited]`

Definition at line 85 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.13.4.44 `ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 92 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.13.4.45 `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()`.

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

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

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

8.13.4.47 `template<class FillT > PointEmitterModel::ParamT` `mappel::PointEmitterModel::make_param (FillT fill) const` `[inherited]`

Definition at line 181 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.13.4.48 `MatT` `mappel::PointEmitterModel::make_param_mat () const` `[inline], [inherited]`

Definition at line 172 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.13.4.49 `template<class FillT > MatT` `mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.13.4.50 `CubeT` `mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline], [inherited]`

Definition at line 176 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.13.4.51 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.13.4.52 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline]`,
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.13.4.53 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.13.4.54 `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.

8.13.4.55 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.13.4.56 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos) [static],[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

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

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

Definition at line 85 of file PointEmitterModel.cpp.

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

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.13.4.59 `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`.

8.13.4.60 `Gauss2DModel::Stencil mappel::Gauss2DModel::make_stencil (const ParamT & theta, bool compute_derivatives = true) const [inline],[inherited]`

Make a new `Model::Stencil` object at `theta`.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular `theta` (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the `mappel::methods` namespace accept a `const Stencil` reference in place of the model parameter.

Throws `mappel::ModelBoundsError` if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`

Parameters

<i>theta</i>	Prameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

Returns

A new [Stencil](#) object ready to compute with

Definition at line 131 of file Gauss2DModel.h.

References [mappel::Gauss2DModel::Stencil::Stencil\(\)](#), and [mappel::PointEmitterModel::theta_in_bounds\(\)](#).

Referenced by [mappel::Gauss2DModel::initial_theta_estimate\(\)](#).

8.13.4.61 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=\(\)](#).

8.13.4.62 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=\(\)](#).

8.13.4.63 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\(\)](#).

8.13.4.64 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](#).

8.13.4.65 `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`.

8.13.4.66 `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::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

8.13.4.67 `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::I()`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

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

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.13.4.69 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const` `[inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.13.4.70 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)` `[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.13.4.71 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)` [inherited]

Definition at line 59 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_u`, `x`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

8.13.4.72 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)` [inherited]

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.13.4.74 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.13.4.75 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.13.4.77 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.13.4.78 `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().

8.13.4.79 `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.

8.13.4.80 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_mean_I, mappel::MCMCAdaptor1D::eta_I, mappel::PointEmitter↵ Model::get_hyperparam_value(), and mappel::MCMCAdaptorBase::sigma_scale.

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

8.13.4.81 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss1DsModel::set_min_sigma().

8.13.4.82 `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().

8.13.4.83 `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.

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

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.13.4.85 `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()`.

8.13.4.86 `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()`.

8.13.4.87 `void mappel::Gauss2DModel::set_psf_sigma (double new_psf_sigma) [inline],[inherited]`

Definition at line 146 of file Gauss2DModel.h.

8.13.4.88 `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`.

8.13.4.89 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.13.4.90 `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`.

8.13.4.91 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::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()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2Dsxymodel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.13.4.94 `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()`.

8.13.5 Member Data Documentation

8.13.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static]`, `[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.13.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static]`, `[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.13.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.13.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.13.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.13.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.13.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.13.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 51 of file Gauss2DModel.h.

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

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

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

Definition at line 23 of file PoissonNoise2DObjective.h.

8.13.5.12 `double mappel::MCMCAdaptor1D::eta_bg =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.13.5.13 `double mappel::MCMCAdaptor1D::eta_l =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.13.5.14 `double mappel::MCMCAdaptor1D::eta_x =0 [protected],[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.13.5.15 `double mappel::MCMCAdaptor2D::eta_y=0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 27 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()`.

8.13.5.16 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

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

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.13.5.17 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.13.5.18 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

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

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

8.13.5.19 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size=512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file `ImageFormat2DBase.h`.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

8.13.5.20 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

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

8.13.5.21 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size =3` `[static]`,
`[inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

Definition at line 37 of file Gauss2DMLE.h.

8.13.5.24 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim =2` `[static]`,
`[inherited]`

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.13.5.27 IdxT mappel::MCMCAdaptorBase::num_phases [protected], [inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_num_phases(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor2Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor2D::sample_mcmc_candidate(), mappel::MCMCAdaptor1Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor1D::sample_mcmc_candidate(), and mappel::MCMCAdaptorBase::set_mcmc_num_phases().

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

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get_hyperparam_index(), mappel::PointEmitterModel::get_hyperparam_names(), mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_hyperparams(), mappel::Gauss1DsModel::get_max_sigma(), mappel::Gauss1DsModel::get_min_sigma(), mappel::PointEmitterModel::get_param_names(), mappel::PointEmitterModel::get_prior(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::has_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename_hyperparam(), mappel::PointEmitterModel::sample_prior(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_hyperparam_names(), mappel::PointEmitterModel::set_hyperparam_value(), mappel::PointEmitterModel::set_hyperparams(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::PointEmitterModel::set_param_names(), mappel::PointEmitterModel::set_prior(), and mappel::PointEmitterModel::set_ubound().

8.13.5.29 const StringVecT mappel::Gauss2DModel::prior_types [static], [inherited]

Initial value:

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

Definition at line 50 of file Gauss2DModel.h.

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

8.13.5.30 VecT mappel::Gauss2DModel::psf_sigma [protected], [inherited]

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 104 of file Gauss2DModel.h.

Referenced by mappel::cgauss_compute_estimate(), mappel::cgauss_compute_estimate_debug(), mappel::cgauss_heuristic_compute_estimate(), mappel::Gauss2DModel::Stencil::compute_derivatives(), mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DModel::get_psf_sigma(), mappel::Gauss2DModel::make_internal_1Dsum_estimator(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::pixel_grad2(), mappel::Gauss2DModel::pixel_hess(), mappel::Gauss2DModel::pixel_hess_update(), mappel::Gauss2DModel::set_psf_sigma(), mappel::Gauss2DModel::Stencil::Stencil(), and mappel::Gauss2DModel::update_internal_1Dsum_estimators().

8.13.5.31 `double mappel::MCMCAdaptorBase::sigma_scale` [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.13.5.32 `ImageSizeT mappel::ImageFormat2DBase::size` [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by `mappel::cgauss_compute_estimate()`, `mappel::cgauss_compute_estimate_debug()`, `mappel::cgauss_heuristic_compute_estimate()`, `mappel::ImageFormat2DBase::check_image_shape()`, `mappel::Gauss2DModel::Stencil::compute_derivatives()`, `mappel::Gauss2DsModel::Stencil::compute_derivatives()`, `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::methods::expected_information()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsMAP::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()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.13.5.34 Gauss1DSumModelT mappel::Gauss2DModel::x_model [protected],[inherited]

X-model fits 2D images X-axis (column sum)

Definition at line 105 of file Gauss2DModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_x(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::set_psf_sigma(), mappel::Gauss2DModel::set_size(), and mappel::Gauss2DModel::update_internal_1Dsum_estimators().

8.13.5.35 Gauss1DSumModelT mappel::Gauss2DModel::y_model [protected],[inherited]

Y-model fits 2D images Y-axis (row sum)

Definition at line 106 of file Gauss2DModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DModel::operator=(), mappel::Gauss2DModel::set_psf_sigma(), mappel::Gauss2DModel::set_size(), and mappel::Gauss2DModel::update_internal_1Dsum_estimators().

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

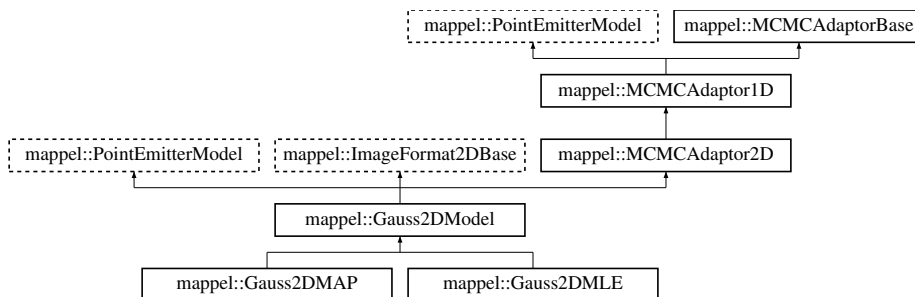
- [Gauss2DMLE.h](#)
- [Gauss2DMLE.cpp](#)

8.14 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)
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)
- `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)
- [ParamT](#) [sample_prior](#) ()
- 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)
- 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_xpos, double pos_ypos, double sigma_pos, double kappa_pos)
 - static prior_hessian::ScaledSymmetricBetaDist `make_prior_component_position_beta` (`IdxT` size, double pos_xpos, double pos_ypos, double beta_pos, double kappa_pos)
 - static prior_hessian::TruncatedGammaDist `make_prior_component_intensity` (double mean_pos, double sigma_pos, double kappa_pos)
 - static prior_hessian::TruncatedParetoDist `make_prior_component_sigma` (double min_sigma, double max_sigma, double alpha_sigma)
 - static void `set_rng_seed` (`RngSeedT` seed)
 - static `ParallelRngManagerT` & `get_rng_manager` ()
 - static `ParallelRngGeneratorT` & `get_rng_generator` ()
 - static void `check_size` (const `ImageSizeT` &size_)
- Check the size argument for the model.*

Static Public Attributes

- static const `StringVecT` `prior_types`
- static const std::string `DefaultPriorType` = "Normal"
- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1
- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2
- static const `ImageCoordT` `num_dim` =2
- static const `ImageCoordT` `global_min_size` =3
- static const `ImageCoordT` `global_max_size` =512
- static const double `global_default_mcmc_sigma_scale` = 0.05
- static const double `global_max_mcmc_sigma_scale` = 0.5

Protected Member Functions

- [Gauss2DModel](#) (const [ImageSizeT](#) &size, const [VecT](#) &psf_sigma)
- [Gauss2DModel](#) (const [Gauss2DModel](#) &o)
- [Gauss2DModel](#) ([Gauss2DModel](#) &&o)
- [Gauss2DModel](#) & operator= (const [Gauss2DModel](#) &o)
- [Gauss2DModel](#) & operator= ([Gauss2DModel](#) &&o)
- void [update_internal_1Dsum_estimators](#) ()
- void [set_mcmc_num_phases](#) ([IdxT](#) num_phases)

Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make_internal_1Dsum_estimator](#) ([IdxT](#) dim, const [ImageSizeT](#) &size, const [VecT](#) &psf_sigma, const [CompositeDist](#) &prior)

Protected Attributes

- [VecT](#) psf_sigma
- [Gauss1DSumModelT](#) x_model
- [Gauss1DSumModelT](#) y_model
- [CompositeDist](#) prior
- [IdxT](#) num_params
- [IdxT](#) num_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double eta_y =0
- double eta_x =0
- double eta_l =0
- double eta_bg =0
- [IdxT](#) num_phases
- double sigma_scale

8.14.1 Detailed Description

A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma.

Definition at line 20 of file [Gauss2DModel.h](#).

8.14.2 Member Typedef Documentation

8.14.2.1 using mappel::Gauss2DModel::Gauss1DSumModelT = Gauss1DMP

Definition at line 23 of file [Gauss2DModel.h](#).

8.14.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` `[inherited]`

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

8.14.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

8.14.2.4 `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.

8.14.2.5 `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.

8.14.2.6 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
`[inherited]`

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

8.14.2.7 `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.

8.14.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.14.2.9 `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.

8.14.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.14.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.14.2.12 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.14.2.13 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.14.2.14 `using mappel::Gauss2DModel::StencilVecT = std::vector<Stencil>`

Definition at line 47 of file Gauss2DModel.h.

8.14.3 Constructor & Destructor Documentation

8.14.3.1 `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()`.

8.14.3.2 `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`.

8.14.3.3 `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`.

8.14.4 Member Function Documentation

8.14.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const` `[inherited]`

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.14.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` `[inherited]`

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.14.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` `[inherited]`

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.14.4.4 `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`.

8.14.4.5 `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.

8.14.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.14.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.14.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and set_psf_sigma().

8.14.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.14.4.10 `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().

8.14.4.11 `Gauss1DSumModelT mappel::Gauss2DModel::debug_internal_sum_model_x () const` `[inline]`

Definition at line 89 of file Gauss2DModel.h.

References `x_model`.

8.14.4.12 `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`.

8.14.4.13 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.14.4.15 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.14.4.16 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.14.4.17 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 106 of file ImageFormat2DBase.h.

8.14.4.18 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.14.4.19 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.14.4.20 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.14.4.21 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`, `[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.14.4.22 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`, `[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.14.4.23 `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().

8.14.4.24 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline]`, `[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_←
estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.14.4.27 const VecT & mappel::Gauss2DModel::get_psf_sigma () const [inline]

Definition at line 142 of file Gauss2DModel.h.

References psf_sigma.

Referenced by get_stats().

8.14.4.28 double mappel::Gauss2DModel::get_psf_sigma (IdxT idx) const

Definition at line 132 of file Gauss2DModel.cpp.

References DefaultPriorType, prior_types, and psf_sigma.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.14.4.31 const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const [inline],[inherited]

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.14.4.32 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT *idx*) const
[*inherited*]

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

8.14.4.33 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & *stack*) const [inline], [*inherited*]

Definition at line 99 of file ImageFormat2DBase.h.

8.14.4.34 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().

8.14.4.35 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline], [*inherited*]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.14.4.36 bool mappel::PointEmitterModel::has_hyperparam (const std::string & *name*) const [inline], [*inherited*]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.14.4.37 Gauss2DModel::Stencil mappel::Gauss2DModel::initial_theta_estimate (const ImageT & *im*) [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().

8.14.4.38 Gauss2DModel::Stencil `mappel::Gauss2DModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init) [inline]`

Definition at line 201 of file Gauss2DModel.h.

References `mappel::PointEmitterModel::DefaultSeperableInitEstimator`, and `initial_theta_estimate()`.

8.14.4.39 Gauss2DModel::Stencil `mappel::Gauss2DModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init, const std::string & estimator)`

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::PointEmitterModel::theta_in_bounds()`, `mappel::PointEmitterModel::ubound`, `x_model`, and `y_model`.

8.14.4.40 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()`.

8.14.4.41 CompositeDist `mappel::Gauss2DModel::make_default_prior_beta_position (const ImageSizeT & size) [static]`

Definition at line 171 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `make_default_prior()`.

8.14.4.42 CompositeDist `mappel::Gauss2DModel::make_default_prior_normal_position (const ImageSizeT & size) [static]`

Definition at line 182 of file Gauss2DModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `make_default_prior()`.

8.14.4.43 ImageFormat2DBase::ImageT `mappel::ImageFormat2DBase::make_image () const [inline], [inherited]`

Definition at line 85 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.14.4.44 ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT *n*) const
[inline], [inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.14.4.45 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().

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

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(),
initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_↵
_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), pixel_↵
hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.14.4.47 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT *fill*) const
[inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.14.4.48 MatT mappel::PointEmitterModel::make_param_mat () const [inline], [inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.14.4.49 template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT *fill*) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.14.4.50 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.14.4.51 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.14.4.52 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

8.14.4.53 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.14.4.54 `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`.

8.14.4.55 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

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

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

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

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.14.4.59 **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`.

8.14.4.60 **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()`.

8.14.4.61 `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=()`.

8.14.4.62 `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`.

8.14.4.63 `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()`.

8.14.4.64 `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::I()`, `psf_sigma`, `mappel::Gauss2DModel::Stencil::X`, and `mappel::Gauss2DModel::Stencil::Y`.

8.14.4.65 `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::I(), psf_sigma, mappel::Gauss2DModel::Stencil::X, and mappel::Gauss2DModel::Stencil::Y.

8.14.4.66 `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::I(), mappel::PointEmitterModel::make_param(), pixel_grad(), psf_sigma, mappel::Gauss2DModel::Stencil::X, and mappel::Gauss2DModel::Stencil::Y.

8.14.4.67 `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::I(), mappel::Gauss2DModel::Stencil::X, and mappel::Gauss2DModel::Stencil::Y.

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

Definition at line 276 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

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

8.14.4.69 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const` `[inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::make_param_stack(), and mappel::PointEmitterModel::reflected_theta().

8.14.4.70 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)` `[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.14.4.71 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0) [inherited]`

Definition at line 59 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::MCMCAdaptor1D::eta_u`, `mappel::MCMCAdaptor2D::eta_y`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

8.14.4.72 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng) [inherited]`

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.14.4.74 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1) [inherited]`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.14.4.75 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_) [inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.14.4.77 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.14.4.78 `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().

8.14.4.79 `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.

8.14.4.80 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_mean_I, mappel::MCMCAdaptor1D::eta_I, mappel::PointEmitterModel::get_hyperparam_value(), and mappel::MCMCAdaptorBase::sigma_scale.

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

8.14.4.81 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss1DsModel::set_min_sigma().

8.14.4.82 `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().

8.14.4.83 `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.

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

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.14.4.85 `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().

8.14.4.86 `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().

8.14.4.87 `void mappel::Gauss2DModel::set_psf_sigma (double new_psf_sigma) [inline]`

Definition at line 146 of file Gauss2DModel.h.

8.14.4.88 `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.

8.14.4.89 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.14.4.90 `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::Image←Format2DBase::size, x_model, and y_model.

8.14.4.91 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds_epsilon, mappel::PointEmitterModel::lbound, mappel::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().

8.14.4.92 **bool** mappel::PointEmitterModel::theta_in_bounds (const ParamT & *theta*) const [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by 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().

8.14.4.93 **BoolVecT** mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & *theta*) const [inherited]

Definition at line 297 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

8.14.4.94 **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().

8.14.5 Member Data Documentation

8.14.5.1 **const double** mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static],[inherited]

Distance from the boundary to constrain in bound_theta and bounded_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss2DsModel::set_max_sigma_ratio(), and mappel::PointEmitterModel::set_ubound().

8.14.5.2 **const double** mappel::PointEmitterModel::default_alpha_sigma = 2 [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.14.5.3 **const double** mappel::PointEmitterModel::default_beta_pos = 3 [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.14.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.14.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.14.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.14.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.14.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 51 of file Gauss2DModel.h.

Referenced by `get_psf_sigma()`.

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

8.14.5.11 `double mappel::MCMCAdaptor1D::eta_bg = 0 [protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.14.5.12 `double mappel::MCMCAdaptor1D::eta_l = 0 [protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.14.5.13 `double mappel::MCMCAdaptor1D::eta_x = 0 [protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.14.5.14 `double mappel::MCMCAdaptor2D::eta_y = 0 [protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 27 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()`.

8.14.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` [static], [inherited]

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.14.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` [static], [inherited]

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.14.5.17 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` [static], [inherited]

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

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

8.14.5.18 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` [static], [inherited]

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

8.14.5.19 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` [static], [inherited]

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

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

8.14.5.20 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` [static], [inherited]

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_lbound(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DsModel::initial_theta_estimate(), initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.14.5.22 const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim=2 [static], [inherited]

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::get_stats().

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::get_num_params(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DModel::initial_theta_estimate(), initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::make_param(), mappel::PointEmitterModel::make_param_mat(), mappel::PointEmitterModel::make_param_mat_stack(), mappel::PointEmitterModel::make_param_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.14.5.25 IdxT mappel::MCMCAdaptorBase::num_phases [protected], [inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_num_phases(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor2D::sample_mcmc_candidate(), mappel::MCMCAdaptor1Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor2Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor1D::sample_mcmc_candidate(), and mappel::MCMCAdaptorBase::set_mcmc_num_phases().

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

Definition at line 145 of file PointEmitterModel.h.

Referenced by debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get_hyperparam_index(), mappel::PointEmitterModel::get_hyperparam_names(), mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_hyperparams(), mappel::Gauss1DsModel::get_max_sigma(), mappel::Gauss1DsModel::get_min_sigma(), mappel::PointEmitterModel::get_param_names(), mappel::PointEmitterModel::get_prior(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::has_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename_hyperparam(), mappel::PointEmitterModel::sample_prior(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_hyperparam_names(), mappel::PointEmitterModel::set_hyperparam_value(), mappel::PointEmitterModel::set_hyperparams(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::PointEmitterModel::set_param_names(), mappel::PointEmitterModel::set_prior(), and mappel::PointEmitterModel::set_ubound().

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

Initial value:

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

Definition at line 50 of file Gauss2DModel.h.

Referenced by get_psf_sigma().

8.14.5.28 VecT mappel::Gauss2DModel::psf_sigma [protected]

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 104 of file Gauss2DModel.h.

Referenced by mappel::cgauss_compute_estimate(), mappel::cgauss_compute_estimate_debug(), mappel::cgauss_heuristic_compute_estimate(), mappel::Gauss2DModel::Stencil::compute_derivatives(), debug_internal_sum_model_y(), Gauss2DModel(), get_psf_sigma(), make_internal_1Dsum_estimator(), operator=(), pixel_grad2(), pixel_hess(), pixel_hess_update(), set_psf_sigma(), mappel::Gauss2DModel::Stencil::Stencil(), and update_internal_1Dsum_estimators().

8.14.5.29 double mappel::MCMCAdaptorBase::sigma_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_sigma_scale(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling(), and mappel::MCMCAdaptorBase::set_mcmc_sigma_scale().

8.14.5.30 `ImageSizeT mappel::ImageFormat2DBase::size` [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file `ImageFormat2DBase.h`.

Referenced by `mappel::cgauss_compute_estimate()`, `mappel::cgauss_compute_estimate_debug()`, `mappel::cgauss_←_heuristic_compute_estimate()`, `mappel::ImageFormat2DBase::check_image_shape()`, `mappel::Gauss2DModel::←Stencil::compute_derivatives()`, `mappel::Gauss2DsModel::Stencil::compute_derivatives()`, `debug_internal_sum_←model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::methods::expected_information()`, `Gauss2DModel()`, `mappel::Gauss2DsMAP::Gauss2DsMAP()`, `mappel::Gauss2DsMLE::Gauss2DsMLE()`, `mappel::←Gauss2DsModel::Gauss2DsModel()`, `mappel::ImageFormat2DBase::get_num_pixels()`, `mappel::ImageFormat2D←Base::get_size()`, `mappel::ImageFormat2DBase::get_stats()`, `mappel::methods::likelihood::grad()`, `mappel::methods←::likelihood::grad2()`, `mappel::methods::likelihood::debug::grad_components()`, `mappel::methods::likelihood::hessian()`, `mappel::methods::likelihood::debug::hessian_components()`, `initial_theta_estimate()`, `mappel::Gauss2DsModel←::initial_theta_estimate()`, `mappel::methods::likelihood::llh()`, `mappel::methods::likelihood::debug::llh_components()`, `make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `make_default_←prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::ImageFormat2←DBase::make_image()`, `mappel::ImageFormat2DBase::make_image_stack()`, `make_internal_1Dsum_estimator()`, `mappel::Gauss2DsModel::make_internal_1Dsum_estimator()`, `make_prior_beta_position()`, `mappel::Gauss2Ds←Model::make_prior_beta_position()`, `make_prior_normal_position()`, `mappel::Gauss2DsModel::make_prior_normal←_position()`, `mappel::methods::model_image()`, `mappel::ImageFormat2DBase::operator=()`, `mappel::methods←::likelihood::rllh()`, `mappel::methods::likelihood::debug::rllh_components()`, `mappel::ImageFormat2DBase::set_size()`, `set_size()`, `mappel::Gauss2DsModel::set_size()`, `mappel::methods::simulate_image()`, `mappel::methods::simulate_←image_from_model()`, `mappel::Gauss2DModel::Stencil::Stencil()`, `mappel::Gauss2DsModel::Stencil::Stencil()`, `update←_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

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

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::←PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_←estimate()`, `initial_theta_estimate()`, `mappel::Gauss2DsModel::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()`.

8.14.5.32 `Gauss1DSumModelT mappel::Gauss2DModel::x_model` [protected]

X-model fits 2D images X-axis (column sum)

Definition at line 105 of file `Gauss2DModel.h`.

Referenced by `debug_internal_sum_model_x()`, `initial_theta_estimate()`, `operator=()`, `set_psf_sigma()`, `set_size()`, and `update_internal_1Dsum_estimators()`.

8.14.5.33 Gauss1DSumModelT mappel::Gauss2DModel::y_model [protected]

Y-model fits 2D images Y-axis (row sum)

Definition at line 106 of file Gauss2DModel.h.

Referenced by debug_internal_sum_model_y(), Gauss2DModel(), initial_theta_estimate(), operator=(), set_psf_←sigma(), set_size(), and update_internal_1Dsum_estimators().

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

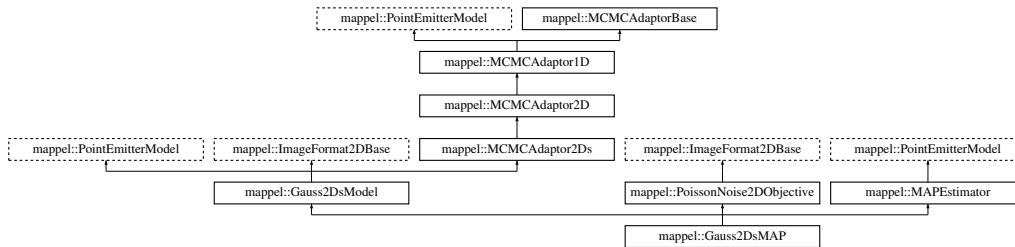
- [Gauss2DModel.h](#)
- [Gauss2DModel.cpp](#)

8.15 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)
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)
- [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 \(RngT &rng\)](#)
- [ParamT sample_prior \(\)](#)
- [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 \(IdxT sample_index, ParamT &candidate, double step_scale=1.0\)](#)
- [void set_intensity_mcmc_sampling \(double eta_l=-1\)](#)
- [void set_background_mcmc_sampling \(double eta_bg=-1\)](#)
- [void set_mcmc_sigma_scale \(double scale\)](#)
- [double get_mcmc_sigma_scale \(\)](#) const
- [IdxT get_mcmc_num_phases \(\)](#) const

Static Public Member Functions

- static CompositeDist [make_default_prior](#) (const [ImageSizeT](#) &size, double max_sigma_ratio, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) (const [ImageSizeT](#) &size, double max_sigma_ratio)
- static CompositeDist [make_default_prior_normal_position](#) (const [ImageSizeT](#) &size, double max_sigma_ratio)
- static CompositeDist [make_prior_beta_position](#) (const [ImageSizeT](#) &size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma)
- static CompositeDist [make_prior_normal_position](#) (const [ImageSizeT](#) &size, double sigma_xpos, double sigma_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) ([IdxT](#) size, double pos_sigma=default_sigma_pos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_beta=default_beta_pos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=default_mean_l, double kappa=default_intensity_kappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_sigma, double alpha=default_alpha_sigma)
- static void [set_rng_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()
- static void [check_size](#) (const [ImageSizeT](#) &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) =2
- static const [ImageCoordT](#) [global_min_size](#) =3
- static const [ImageCoordT](#) [global_max_size](#) =512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [update_internal_1Dsum_estimators](#) ()
- void [set_mcmc_num_phases](#) (IdxT num_phases)

Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make_internal_1Dsum_estimator](#) (IdxT dim, const [ImageSizeT](#) &size, const [VecT](#) &min_sigma, const [VecT](#) &max_sigma, const [CompositeDist](#) &prior)
- static double [compute_max_sigma_ratio](#) (const [VecT](#) &min_sigma, const [VecT](#) &max_sigma)

Protected Attributes

- [VecT](#) min_sigma
- [Gauss1DSumModelT](#) x_model
- [Gauss1DSumModelT](#) y_model
- [CompositeDist](#) prior
- [IdxT](#) num_params
- [IdxT](#) num_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double eta_sigma =0
- double eta_y =0
- double eta_x =0
- double eta_l =0
- double eta_bg =0
- [IdxT](#) num_phases
- double sigma_scale

8.15.1 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.

8.15.2 Member Typedef Documentation**8.15.2.1 using mappel::Gauss2DsModel::Gauss1DSumModelT = Gauss1DsMAP [inherited]**

Definition at line 26 of file Gauss2DsModel.h.

8.15.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` `[inherited]`

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

8.15.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

8.15.2.4 `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.

8.15.2.5 `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.

8.15.2.6 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
`[inherited]`

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

8.15.2.7 `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.

8.15.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.15.2.9 `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.

8.15.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.15.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.15.2.12 `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.

8.15.2.13 `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.

8.15.2.14 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.15.2.15 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.15.2.16 `using mappel::Gauss2DsModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 55 of file Gauss2DsModel.h.

8.15.3 Constructor & Destructor Documentation

8.15.3.1 `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`.

8.15.3.2 `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`.

8.15.3.3 `mappel::Gauss2DsMAP::Gauss2DsMAP (const ImageSizeT & size, const VecT & min_sigma, CompositeDist && prior)`

Definition at line 19 of file Gauss2DsMAP.cpp.

8.15.3.4 `mappel::Gauss2DsMAP::Gauss2DsMAP (const ImageSizeT & size, const VecT & min_sigma, const CompositeDist & prior)`

Definition at line 27 of file Gauss2DsMAP.cpp.

8.15.3.5 `mappel::Gauss2DsMAP::Gauss2DsMAP (const Gauss2DsMAP & o)`

Definition at line 35 of file Gauss2DsMAP.cpp.

8.15.3.6 `mappel::Gauss2DsMAP::Gauss2DsMAP (Gauss2DsMAP && o)`

Definition at line 43 of file Gauss2DsMAP.cpp.

8.15.4 Member Function Documentation

8.15.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const`
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.15.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 265 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.15.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.15.4.4 `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`.

8.15.4.5 `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`.

8.15.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.15.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::num_params`.

8.15.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` `[inherited]`

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.15.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.15.4.10 `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()`.

8.15.4.11 `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()`.

8.15.4.12 `Gauss1DSumModelT mappel::Gauss2DsModel::debug_internal_sum_model_x () const` `[inline],[inherited]`

Definition at line 104 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::x_model`.

8.15.4.13 `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`.

8.15.4.14 `int` `mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

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

References `mappel::PointEmitterModel::prior`.

8.15.4.16 `double` `mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.15.4.17 `PointEmitterModel::ParamT` `mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.15.4.18 `ImageFormat2DBase::ImageT` `mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 106 of file `ImageFormat2DBase.h`.

8.15.4.19 `const PointEmitterModel::ParamT &` `mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.15.4.20 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().

8.15.4.21 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().

8.15.4.22 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().

8.15.4.23 IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const [inherited]

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.15.4.24 double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const [inherited]

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.15.4.25 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().

8.15.4.26 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.

8.15.4.27 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.15.4.28 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.15.4.29 `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().

8.15.4.30 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel←
::update_internal_1Dsum_estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.15.4.35 `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const [inline],[inherited]`

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.15.4.36 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT idx) const [inherited]`

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

8.15.4.37 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]`

Definition at line 99 of file ImageFormat2DBase.h.

8.15.4.38 `StatsT mappel::Gauss2DsModel::get_stats () const [inherited]`

Definition at line 341 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().

8.15.4.39 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.15.4.40 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline],[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.15.4.41 Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & im) [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().

8.15.4.42 Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init) [inline], [inherited]

Definition at line 231 of file Gauss2DsModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::Gauss2DsModel::initial_theta_estimate().

8.15.4.43 Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init, const std::string & estimator) [inherited]

Definition at line 385 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::PointEmitterModel::theta_in_bounds(), mappel::PointEmitterModel::ubound, mappel::Gauss2DsModel::x_model, and mappel::Gauss2DsModel::y_model.

8.15.4.44 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().

8.15.4.45 CompositeDist mappel::Gauss2DsModel::make_default_prior_beta_position (const ImageSizeT & size, double max_sigma_ratio) [static], [inherited]

Definition at line 230 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::default_pixel_mean_bg, mappel::PointEmitterModel::make_prior_component_intensity(), mappel::PointEmitterModel::make_prior_component_position_beta(), mappel::PointEmitterModel::make_prior_component_sigma(), and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DsModel::make_default_prior().

8.15.4.46 `CompositeDist mappel::Gauss2DsModel::make_default_prior_normal_position (const ImageSizeT & size, double max_sigma_ratio) [static], [inherited]`

Definition at line 243 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

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

8.15.4.47 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const [inline], [inherited]`

Definition at line 85 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.15.4.48 `ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT n) const [inline], [inherited]`

Definition at line 92 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.15.4.49 `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::Gauss2DsModel()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

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

Definition at line 164 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

8.15.4.51 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const`
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.15.4.52 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.15.4.53 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.15.4.54 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.15.4.55 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.15.4.56 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.15.4.57 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.15.4.58 **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 255 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.

8.15.4.59 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa) [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

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

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), and mappel::Gauss2DsModel::make_prior_beta_position().

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

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.15.4.63 `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 271 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`.

8.15.4.64 `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()`.

8.15.4.65 Gauss2DsMAP & mappel::Gauss2DsMAP::operator= (const Gauss2DsMAP & o)

Definition at line 51 of file Gauss2DsMAP.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::Image↵
Format2DBase::operator=(), mappel::Gauss2DsModel::operator=(), and mappel::PointEmitterModel::operator=().

8.15.4.66 Gauss2DsMAP & mappel::Gauss2DsMAP::operator= (Gauss2DsMAP && o)

Definition at line 62 of file Gauss2DsMAP.cpp.

References mappel::MAPEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::Image↵
Format2DBase::operator=(), mappel::Gauss2DsModel::operator=(), and mappel::PointEmitterModel::operator=().

8.15.4.67 void mappel::Gauss2DsModel::pixel_grad (int i, int j, const Stencil & s, ParamT & pgrad) const [inline],
[inherited]

Definition at line 180 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵
Model::Stencil::DY, mappel::Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::I(), mappel::Gauss2↵
DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

Referenced by mappel::Gauss2DsModel::pixel_hess_update().

8.15.4.68 void mappel::Gauss2DsModel::pixel_grad2 (int i, int j, const Stencil & s, ParamT & pgrad2) const [inline],
[inherited]

Definition at line 192 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2DsModel::Stencil::DXS2, mappel::Gauss2↵
DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::I(), mappel::↵
Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), mappel::Gauss2DsModel::Stencil::X,
and mappel::Gauss2DsModel::Stencil::Y.

8.15.4.69 void mappel::Gauss2DsModel::pixel_hess (int i, int j, const Stencil & s, MatT & hess) const [inline],
[inherited]

Definition at line 204 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵
Model::Stencil::DXS2, mappel::Gauss2DsModel::Stencil::DXSX, mappel::Gauss2DsModel::Stencil::DY, mappel::↵
Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::DYSY,
mappel::Gauss2DsModel::Stencil::I(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil↵
::sigmaY(), mappel::Gauss2DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

8.15.4.70 `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 358 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::Stencil::DX`, `mappel::Gauss2DsModel::Stencil::DXS`, `mappel::Gauss2DsModel::Stencil::DXS2`, `mappel::Gauss2DsModel::Stencil::DXSX`, `mappel::Gauss2DsModel::Stencil::DY`, `mappel::Gauss2DsModel::Stencil::DYS`, `mappel::Gauss2DsModel::Stencil::DYS2`, `mappel::Gauss2DsModel::Stencil::DYSY`, `mappel::Gauss2DsModel::Stencil::I()`, `mappel::PointEmitterModel::make_param()`, `mappel::Gauss2DsModel::pixel_grad()`, `mappel::Gauss2DsModel::Stencil::sigmaX()`, `mappel::Gauss2DsModel::Stencil::sigmaY()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

8.15.4.71 `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::I()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

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

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.15.4.73 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const` `[inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.15.4.74 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)` `[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.15.4.75 `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)` [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`.

8.15.4.76 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)` [inherited]

Definition at line 264 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.15.4.78 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::Point`
`EmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_`
`ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.15.4.79 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.15.4.81 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value) [inline],
[inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.15.4.82 `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().

8.15.4.83 `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.

8.15.4.84 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_mean_I, mappel::MCMCAdaptor1D::eta_I, mappel::PointEmitter←
Model::get_hyperparam_value(), and mappel::MCMCAdaptorBase::sigma_scale.

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

8.15.4.85 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss1DsModel::set_min_sigma().

8.15.4.86 `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().

8.15.4.87 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().

8.15.4.88 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().

8.15.4.89 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.

8.15.4.90 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.

8.15.4.91 void mappel::PointEmitterModel::set_param_names (const StringVecT & *desc*) [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.15.4.92 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().

8.15.4.93 `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()`.

8.15.4.94 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.15.4.95 `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`.

8.15.4.96 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::Point←
EmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make←
_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::Point←
EmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.15.4.99 `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()`.

8.15.5 Member Data Documentation

8.15.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static], [inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.15.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.15.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.15.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static], [inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.15.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` [static], [inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.15.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.15.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.15.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 59 of file Gauss2DsModel.h.

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

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

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

Definition at line 23 of file PoissonNoise2DObjective.h.

8.15.5.12 `double mappel::MCMCAdaptor1D::eta_bg=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.15.5.13 `double mappel::MCMCAdaptor1D::eta_l=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.15.5.14 `double mappel::MCMCAdaptor2Ds::eta_sigma=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 26 of file `MCMCAdaptor2Ds.h`.

Referenced by `mappel::MCMCAdaptor2Ds::get_stats()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor2Ds::operator=()`, and `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`.

8.15.5.15 `double mappel::MCMCAdaptor1D::eta_x=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.15.5.16 `double mappel::MCMCAdaptor2D::eta_y=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 27 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()`.

8.15.5.17 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.15.5.18 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.15.5.19 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

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

8.15.5.20 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

8.15.5.21 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

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

8.15.5.22 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_lbound(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.15.5.24 VecT mappel::Gauss2DsModel::min_sigma [protected],[inherited]

Gaussian PSF in pixels

Definition at line 118 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::compute_max_sigma_ratio(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), Gauss2DsMAP(), mappel::Gauss2DsMLE::Gauss2DsMLE(), mappel::Gauss2DsModel::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().

8.15.5.25 const std::string mappel::Gauss2DsMAP::name [static]

Definition at line 36 of file Gauss2DsMAP.h.

8.15.5.26 const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim =2 [static],[inherited]

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::get_stats().

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::get_num_params(), mappel::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().

8.15.5.29 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

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

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

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

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.15.5.31 `const StringVecT mappel::Gauss2DsModel::prior_types` `[static]`, `[inherited]`

Initial value:

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

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

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

8.15.5.32 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`, `[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.15.5.33 ImageSizeT mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss_compute_estimate(), mappel::cgauss_compute_estimate_debug(), mappel::cgauss_←
_heuristic_compute_estimate(), mappel::ImageFormat2DBase::check_image_shape(), mappel::Gauss2DModel::←
Stencil::compute_derivatives(), mappel::Gauss2DsModel::Stencil::compute_derivatives(), mappel::Gauss2DModel::←
::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::methods::←
::expected_information(), mappel::Gauss2DModel::Gauss2DModel(), Gauss2DsMAP(), mappel::Gauss2DsMLE::←
::Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::ImageFormat2DBase::get_num_pixels(),
mappel::ImageFormat2DBase::get_size(), mappel::ImageFormat2DBase::get_stats(), mappel::methods::likelihood←
::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood::debug::grad_components(), mappel::←
::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian_components(), mappel::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().

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::←
::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta←
_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(),
mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel←
::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::Point←
EmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.15.5.35 Gauss1DSumModelT mappel::Gauss2DsModel::x_model [protected],[inherited]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 119 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::debug_internal_sum_model_x(), mappel::Gauss2DsModel::initial_theta←
_estimate(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set_max_sigma_ratio(), mappel::←
Gauss2DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_size(), and mappel::Gauss2DsModel::update_←
internal_1Dsum_estimators().

8.15.5.36 Gauss1DSumModelT mappel::Gauss2DsModel::y_model [protected],[inherited]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 120 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set_max_sigma_ratio(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_size(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

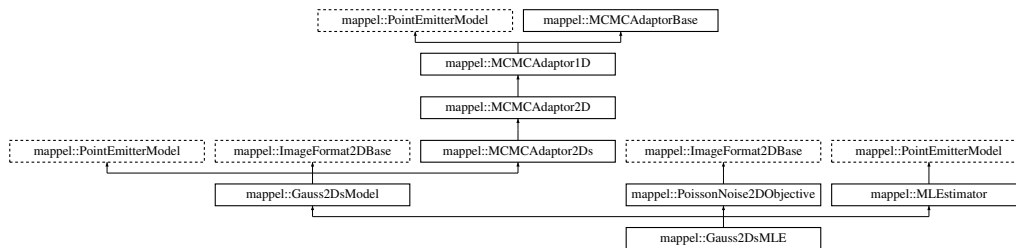
- [Gauss2DsMAP.h](#)
- [Gauss2DsMAP.cpp](#)

8.16 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)
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)
- [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 \(RngT &rng\)](#)
- [ParamT sample_prior \(\)](#)
- [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 \(IdxT sample_index, ParamT &candidate, double step_scale=1.0\)](#)
- [void set_intensity_mcmc_sampling \(double eta_l=-1\)](#)
- [void set_background_mcmc_sampling \(double eta_bg=-1\)](#)
- [void set_mcmc_sigma_scale \(double scale\)](#)
- [double get_mcmc_sigma_scale \(\)](#) const
- [IdxT get_mcmc_num_phases \(\)](#) const

Static Public Member Functions

- static CompositeDist [make_default_prior](#) (const [ImageSizeT](#) &size, double max_sigma_ratio, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) (const [ImageSizeT](#) &size, double max_sigma_ratio)
- static CompositeDist [make_default_prior_normal_position](#) (const [ImageSizeT](#) &size, double max_sigma_ratio)
- static CompositeDist [make_prior_beta_position](#) (const [ImageSizeT](#) &size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma)
- static CompositeDist [make_prior_normal_position](#) (const [ImageSizeT](#) &size, double sigma_xpos, double sigma_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) ([IdxT](#) size, double pos_sigma=default_sigma_pos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_beta=default_beta_pos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=default_mean_l, double kappa=default_intensity_kappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_sigma, double alpha=default_alpha_sigma)
- static void [set_rng_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()
- static void [check_size](#) (const [ImageSizeT](#) &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) =2
- static const [ImageCoordT](#) [global_min_size](#) =3
- static const [ImageCoordT](#) [global_max_size](#) =512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [update_internal_1Dsum_estimators](#) ()
- void [set_mcmc_num_phases](#) (IdxT num_phases)

Static Protected Member Functions

- static [Gauss1DSumModelT](#) [make_internal_1Dsum_estimator](#) (IdxT dim, const [ImageSizeT](#) &size, const [VecT](#) &min_sigma, const [VecT](#) &max_sigma, const [CompositeDist](#) &prior)
- static double [compute_max_sigma_ratio](#) (const [VecT](#) &min_sigma, const [VecT](#) &max_sigma)

Protected Attributes

- [VecT](#) min_sigma
- [Gauss1DSumModelT](#) x_model
- [Gauss1DSumModelT](#) y_model
- [CompositeDist](#) prior
- [IdxT](#) num_params
- [IdxT](#) num_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size
- double [eta_sigma](#) =0
- double [eta_y](#) =0
- double [eta_x](#) =0
- double [eta_l](#) =0
- double [eta_bg](#) =0
- [IdxT](#) num_phases
- double [sigma_scale](#)

8.16.1 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.

8.16.2 Member Typedef Documentation

8.16.2.1 `using mappel::Gauss2DsModel::Gauss1DSumModelT = Gauss1DsMAP` [\[inherited\]](#)

Definition at line 26 of file Gauss2DsModel.h.

8.16.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 24 of file ImageFormat2DBase.h.

8.16.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 25 of file ImageFormat2DBase.h.

8.16.2.4 `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.

8.16.2.5 `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.

8.16.2.6 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
[inherited]

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

8.16.2.7 `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.

8.16.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
[inherited]

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.16.2.9 `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.

8.16.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.16.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.16.2.12 `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.

8.16.2.13 `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.

8.16.2.14 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.16.2.15 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.16.2.16 `using mappel::Gauss2DsModel::StencilVecT = std::vector<Stencil>` `[inherited]`

Definition at line 55 of file Gauss2DsModel.h.

8.16.3 Constructor & Destructor Documentation

8.16.3.1 `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`.

8.16.3.2 `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`.

8.16.3.3 `mappel::Gauss2DsMLE::Gauss2DsMLE (const ImageSizeT & size, const VecT & min_sigma, CompositeDist && prior)`

Definition at line 19 of file Gauss2DsMLE.cpp.

8.16.3.4 `mappel::Gauss2DsMLE::Gauss2DsMLE (const ImageSizeT & size, const VecT & min_sigma, const CompositeDist & prior)`

Definition at line 27 of file Gauss2DsMLE.cpp.

8.16.3.5 `mappel::Gauss2DsMLE::Gauss2DsMLE (const Gauss2DsMLE & o)`

Definition at line 35 of file Gauss2DsMLE.cpp.

8.16.3.6 `mappel::Gauss2DsMLE::Gauss2DsMLE (Gauss2DsMLE && o)`

Definition at line 43 of file Gauss2DsMLE.cpp.

8.16.4 Member Function Documentation

8.16.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const`
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.16.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.16.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.16.4.4 `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`.

8.16.4.5 `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`.

8.16.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.16.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.16.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` `[inherited]`

Definition at line 185 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.16.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` `[inherited]`

Definition at line 197 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.16.4.10 `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()`.

8.16.4.11 `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()`.

8.16.4.12 `Gauss1DSumModelT mappel::Gauss2DsModel::debug_internal_sum_model_x () const` `[inline],[inherited]`

Definition at line 104 of file `Gauss2DsModel.h`.

References `mappel::Gauss2DsModel::x_model`.

8.16.4.13 `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`.

8.16.4.14 `int` `mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

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

References `mappel::PointEmitterModel::prior`.

8.16.4.16 `double` `mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.16.4.17 `PointEmitterModel::ParamT` `mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.16.4.18 `ImageFormat2DBase::ImageT` `mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 106 of file `ImageFormat2DBase.h`.

8.16.4.19 `const PointEmitterModel::ParamT &` `mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.16.4.20 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().

8.16.4.21 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().

8.16.4.22 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().

8.16.4.23 IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const [inherited]

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.16.4.24 double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const [inherited]

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.16.4.25 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().

8.16.4.26 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.

8.16.4.27 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline],[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.16.4.28 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline],[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.16.4.29 `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().

8.16.4.30 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel←
::update_internal_1Dsum_estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.16.4.35 `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const [inline],[inherited]`

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.16.4.36 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT idx) const [inherited]`

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

8.16.4.37 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]`

Definition at line 99 of file ImageFormat2DBase.h.

8.16.4.38 `StatsT mappel::Gauss2DsModel::get_stats () const [inherited]`

Definition at line 341 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().

8.16.4.39 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.16.4.40 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline],[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.16.4.41 Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & im) [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().

8.16.4.42 Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init) [inline], [inherited]

Definition at line 231 of file Gauss2DsModel.h.

References mappel::PointEmitterModel::DefaultSeperableInitEstimator, and mappel::Gauss2DsModel::initial_theta_estimate().

8.16.4.43 Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init, const std::string & estimator) [inherited]

Definition at line 385 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::PointEmitterModel::theta_in_bounds(), mappel::PointEmitterModel::ubound, mappel::Gauss2DsModel::x_model, and mappel::Gauss2DsModel::y_model.

8.16.4.44 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().

8.16.4.45 CompositeDist mappel::Gauss2DsModel::make_default_prior_beta_position (const ImageSizeT & size, double max_sigma_ratio) [static], [inherited]

Definition at line 230 of file Gauss2DsModel.cpp.

References mappel::PointEmitterModel::default_pixel_mean_bg, mappel::PointEmitterModel::make_prior_component_intensity(), mappel::PointEmitterModel::make_prior_component_position_beta(), mappel::PointEmitterModel::make_prior_component_sigma(), and mappel::ImageFormat2DBase::size.

Referenced by mappel::Gauss2DsModel::make_default_prior().

8.16.4.46 `CompositeDist mappel::Gauss2DsModel::make_default_prior_normal_position (const ImageSizeT & size, double max_sigma_ratio) [static], [inherited]`

Definition at line 243 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

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

8.16.4.47 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const [inline], [inherited]`

Definition at line 85 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.16.4.48 `ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT n) const [inline], [inherited]`

Definition at line 92 of file ImageFormat2DBase.h.

References `mappel::ImageFormat2DBase::size`.

8.16.4.49 `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()`.

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

Definition at line 164 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

8.16.4.51 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const`
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.16.4.52 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.16.4.53 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.16.4.54 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.16.4.55 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.16.4.56 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.16.4.57 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.16.4.58 **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 255 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.

8.16.4.59 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa) [static], [inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

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

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), and mappel::Gauss2DsModel::make_prior_beta_position().

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

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.16.4.63 `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 271 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`.

8.16.4.64 `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()`.

8.16.4.65 Gauss2DsMLE & mappel::Gauss2DsMLE::operator= (const Gauss2DsMLE & o)

Definition at line 51 of file Gauss2DsMLE.cpp.

References mappel::MLEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::Image↵
Format2DBase::operator=(), mappel::Gauss2DsModel::operator=(), and mappel::PointEmitterModel::operator=().

8.16.4.66 Gauss2DsMLE & mappel::Gauss2DsMLE::operator= (Gauss2DsMLE && o)

Definition at line 62 of file Gauss2DsMLE.cpp.

References mappel::MLEstimator::operator=(), mappel::PoissonNoise2DObjective::operator=(), mappel::Image↵
Format2DBase::operator=(), mappel::Gauss2DsModel::operator=(), and mappel::PointEmitterModel::operator=().

8.16.4.67 void mappel::Gauss2DsModel::pixel_grad (int i, int j, const Stencil & s, ParamT & pgrad) const [inline],
[inherited]

Definition at line 180 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵
Model::Stencil::DY, mappel::Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::I(), mappel::Gauss2↵
DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

Referenced by mappel::Gauss2DsModel::pixel_hess_update().

8.16.4.68 void mappel::Gauss2DsModel::pixel_grad2 (int i, int j, const Stencil & s, ParamT & pgrad2) const [inline],
[inherited]

Definition at line 192 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2DsModel::Stencil::DXS2, mappel::Gauss2↵
DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::I(), mappel::↵
Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), mappel::Gauss2DsModel::Stencil::X,
and mappel::Gauss2DsModel::Stencil::Y.

8.16.4.69 void mappel::Gauss2DsModel::pixel_hess (int i, int j, const Stencil & s, MatT & hess) const [inline],
[inherited]

Definition at line 204 of file Gauss2DsModel.h.

References mappel::Gauss2DsModel::Stencil::DX, mappel::Gauss2DsModel::Stencil::DXS, mappel::Gauss2Ds↵
Model::Stencil::DXS2, mappel::Gauss2DsModel::Stencil::DXSX, mappel::Gauss2DsModel::Stencil::DY, mappel::↵
Gauss2DsModel::Stencil::DYS, mappel::Gauss2DsModel::Stencil::DYS2, mappel::Gauss2DsModel::Stencil::DYSY,
mappel::Gauss2DsModel::Stencil::I(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil↵
::sigmaY(), mappel::Gauss2DsModel::Stencil::X, and mappel::Gauss2DsModel::Stencil::Y.

8.16.4.70 `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 358 of file Gauss2DsModel.cpp.

References `mappel::Gauss2DsModel::Stencil::DX`, `mappel::Gauss2DsModel::Stencil::DXS`, `mappel::Gauss2DsModel::Stencil::DXS2`, `mappel::Gauss2DsModel::Stencil::DXSX`, `mappel::Gauss2DsModel::Stencil::DY`, `mappel::Gauss2DsModel::Stencil::DYS`, `mappel::Gauss2DsModel::Stencil::DYS2`, `mappel::Gauss2DsModel::Stencil::DYSY`, `mappel::Gauss2DsModel::Stencil::I()`, `mappel::PointEmitterModel::make_param()`, `mappel::Gauss2DsModel::pixel_grad()`, `mappel::Gauss2DsModel::Stencil::sigmaX()`, `mappel::Gauss2DsModel::Stencil::sigmaY()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

8.16.4.71 `double mappel::Gauss2DsModel::pixel_model_value (int i, int j, const Stencil & s) const` `[inline]`, `[inherited]`

Definition at line 173 of file Gauss2DsModel.h.

References `mappel::Gauss2DsModel::Stencil::bg()`, `mappel::Gauss2DsModel::Stencil::I()`, `mappel::Gauss2DsModel::Stencil::X`, and `mappel::Gauss2DsModel::Stencil::Y`.

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

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.16.4.73 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const` `[inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.16.4.74 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)` `[inline]`, `[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.16.4.75 `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)` [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.

8.16.4.76 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)` [inherited]

Definition at line 264 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 268 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior, and mappel::rng_manager.

8.16.4.78 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_pixel_mean_bg, mappel::MCMCAdaptor1D::eta_bg, mappel::Point↵EmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_lbound(), mappel::PointEmitterModel::get_↵ubound(), and mappel::MCMCAdaptorBase::sigma_scale.

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

8.16.4.79 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.16.4.81 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value) [inline],
[inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.16.4.82 `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().

8.16.4.83 `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.

8.16.4.84 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_mean_I, mappel::MCMCAdaptor1D::eta_I, mappel::PointEmitter←
Model::get_hyperparam_value(), and mappel::MCMCAdaptorBase::sigma_scale.

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

8.16.4.85 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss1DsModel::set_min_sigma().

8.16.4.86 `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().

8.16.4.87 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().

8.16.4.88 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().

8.16.4.89 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.

8.16.4.90 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.

8.16.4.91 void mappel::PointEmitterModel::set_param_names (const *StringVecT & desc*) [inline],[inherited]

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.16.4.92 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().

8.16.4.93 `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()`.

8.16.4.94 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.16.4.95 `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`.

8.16.4.96 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::Point← EmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make← _stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::Point← EmitterModel::theta_stack_in_bounds()`.

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

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.16.4.99 `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()`.

8.16.5 Member Data Documentation

8.16.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static], [inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.16.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.16.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static], [inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.16.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static], [inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.16.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` [static], [inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.16.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.16.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.16.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 59 of file Gauss2DsModel.h.

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

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

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

Definition at line 23 of file PoissonNoise2DObjective.h.

8.16.5.12 `double mappel::MCMCAdaptor1D::eta_bg=0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.16.5.13 `double mappel::MCMCAdaptor1D::eta_l=0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.16.5.14 `double mappel::MCMCAdaptor2Ds::eta_sigma=0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 26 of file `MCMCAdaptor2Ds.h`.

Referenced by `mappel::MCMCAdaptor2Ds::get_stats()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor2Ds::operator=()`, and `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`.

8.16.5.15 `double mappel::MCMCAdaptor1D::eta_x=0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.16.5.16 `double mappel::MCMCAdaptor2D::eta_y=0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 27 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()`.

8.16.5.17 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.16.5.18 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.16.5.19 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

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

8.16.5.20 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

8.16.5.21 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

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

8.16.5.22 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_lbound(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.16.5.24 VecT mappel::Gauss2DsModel::min_sigma [protected],[inherited]

Gaussian PSF in pixels

Definition at line 118 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::compute_max_sigma_ratio(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DsMAP::Gauss2DsMAP(), Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss2DsModel::get_min_sigma(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss2DsModel::make_internal_1Dsum_estimator(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

Definition at line 36 of file Gauss2DsMLE.h.

8.16.5.26 const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim =2 [static],[inherited]

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::get_stats().

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::get_num_params(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::make_param(), mappel::PointEmitterModel::make_param_mat(), mappel::PointEmitterModel::make_param_mat_stack(), mappel::PointEmitterModel::make_param_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.16.5.29 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

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

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

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

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.16.5.31 `const StringVecT mappel::Gauss2DsModel::prior_types` `[static]`, `[inherited]`

Initial value:

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

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

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

8.16.5.32 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`, `[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.16.5.33 ImageSizeT mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss_compute_estimate(), mappel::cgauss_compute_estimate_debug(), mappel::cgauss_←
_heuristic_compute_estimate(), mappel::ImageFormat2DBase::check_image_shape(), mappel::Gauss2DModel::←
Stencil::compute_derivatives(), mappel::Gauss2DsModel::Stencil::compute_derivatives(), mappel::Gauss2DModel::←
::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::methods::←
::expected_information(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsMAP::Gauss2DsMAP(),
Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::ImageFormat2DBase::get_num_pixels(),
mappel::ImageFormat2DBase::get_size(), mappel::ImageFormat2DBase::get_stats(), mappel::methods::likelihood←
::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood::debug::grad_components(), mappel←
::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian_components(), mappel::Gauss2D←
Model::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::methods::likelihood←
::llh(), mappel::methods::likelihood::debug::llh_components(), mappel::Gauss2DModel::make_default_prior_beta_←
position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_←
prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::ImageFormat2D←
Base::make_image(), mappel::ImageFormat2DBase::make_image_stack(), mappel::Gauss2DModel::make_internal←
_1Dsum_estimator(), mappel::Gauss2DsModel::make_internal_1Dsum_estimator(), mappel::Gauss2DModel::make_←
_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_←
prior_normal_position(), mappel::Gauss2DsModel::make_prior_normal_position(), mappel::methods::model_image(),
mappel::ImageFormat2DBase::operator=(), mappel::methods::likelihood::rllh(), mappel::methods::likelihood::debug←
::rllh_components(), mappel::ImageFormat2DBase::set_size(), mappel::Gauss2DModel::set_size(), mappel::Gauss2←
DsModel::set_size(), mappel::methods::simulate_image(), mappel::methods::simulate_image_from_model(), mappel←
::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2DsModel::Stencil::Stencil(), mappel::Gauss2DModel::update_←
internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel←
::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta←
_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(),
mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel←
::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::Point←
EmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.16.5.35 Gauss1DSumModelT mappel::Gauss2DsModel::x_model [protected],[inherited]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 119 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::debug_internal_sum_model_x(), mappel::Gauss2DsModel::initial_theta←
_estimate(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set_max_sigma_ratio(), mappel::←
Gauss2DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_size(), and mappel::Gauss2DsModel::update_←
internal_1Dsum_estimators().

8.16.5.36 Gauss1DSumModelT mappel::Gauss2DsModel::y_model [protected],[inherited]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 120 of file Gauss2DsModel.h.

Referenced by mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2DsModel::set_max_sigma_ratio(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_size(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

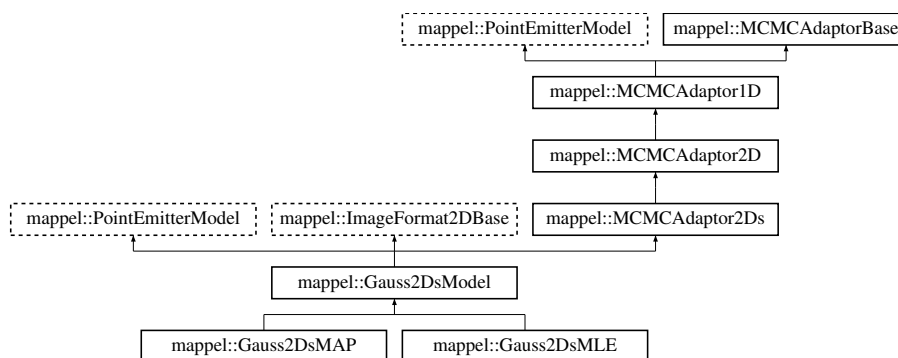
- [Gauss2DsMLE.h](#)
- [Gauss2DsMLE.cpp](#)

8.17 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)
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)
- [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)
- [ParamT sample_prior](#) ()
- 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)
- void [set_intensity_mcmc_sampling](#) (double eta_l=-1)
- void [set_background_mcmc_sampling](#) (double eta_bg=-1)
- void [set_mcmc_sigma_scale](#) (double scale)
- double [get_mcmc_sigma_scale](#) () const
- IdxT [get_mcmc_num_phases](#) () const

Static Public Member Functions

- static CompositeDist [make_default_prior](#) (const ImageSizeT &size, double max_sigma_ratio, const std::string &prior_type)
- static CompositeDist [make_default_prior_beta_position](#) (const ImageSizeT &size, double max_sigma_ratio)
- static CompositeDist [make_default_prior_normal_position](#) (const ImageSizeT &size, double max_sigma_ratio)
- static CompositeDist [make_prior_beta_position](#) (const ImageSizeT &size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma)
- static CompositeDist [make_prior_normal_position](#) (const ImageSizeT &size, double sigma_xpos, double sigma_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma)
- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) (IdxT size, double pos_sigma=default_sigma_pos)
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_beta=default_beta_pos)
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=default_mean_l, double kappa=default_intensity_kappa)
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_sigma, double alpha=default_alpha_sigma)
- static void [set_rng_seed](#) (RngSeedT seed)
- static ParallelRngManagerT & [get_rng_manager](#) ()
- static ParallelRngGeneratorT & [get_rng_generator](#) ()
- static void [check_size](#) (const ImageSizeT &size_)
Check the size argument for the model.

Static Public Attributes

- static const StringVecT [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1

- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2
- static const `ImageCoordT` `num_dim` =2
- static const `ImageCoordT` `global_min_size` =3
- static const `ImageCoordT` `global_max_size` =512
- static const double `global_default_mcmc_sigma_scale` = 0.05
- static const double `global_max_mcmc_sigma_scale` = 0.5

Protected Member Functions

- `Gauss2DsModel` (const `ImageSizeT` &`size`, const `VecT` &`min_sigma`, const `VecT` &`max_sigma`)
- `Gauss2DsModel` (const `Gauss2DsModel` &`o`)
- `Gauss2DsModel` (`Gauss2DsModel` &&`o`)
- `Gauss2DsModel` & `operator=` (const `Gauss2DsModel` &`o`)
- `Gauss2DsModel` & `operator=` (`Gauss2DsModel` &&`o`)
- void `update_internal_1Dsum_estimators` ()
- void `set_mcmc_num_phases` (`IdxT` `num_phases`)

Static Protected Member Functions

- static `Gauss1DSumModelT` `make_internal_1Dsum_estimator` (`IdxT` `dim`, const `ImageSizeT` &`size`, const `VecT` &`min_sigma`, const `VecT` &`max_sigma`, const `CompositeDist` &`prior`)
- static double `compute_max_sigma_ratio` (const `VecT` &`min_sigma`, const `VecT` &`max_sigma`)

Protected Attributes

- `VecT` `min_sigma`
- `Gauss1DSumModelT` `x_model`
- `Gauss1DSumModelT` `y_model`
- `CompositeDist` `prior`
- `IdxT` `num_params`
- `IdxT` `num_hyperparams`
- `ParamT` `lbound`
- `ParamT` `ubound`
- `ImageSizeT` `size`
- double `eta_sigma` =0
- double `eta_y` =0
- double `eta_x` =0
- double `eta_l` =0
- double `eta_bg` =0
- `IdxT` `num_phases`
- double `sigma_scale`

8.17.1 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`.

8.17.2 Member Typedef Documentation

8.17.2.1 `using mappel::Gauss2DsModel::Gauss1DSumModelT = Gauss1DsMAP`

Definition at line 26 of file `Gauss2DsModel.h`.

8.17.2.2 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` `[inherited]`

Image size coordinate storage type

Definition at line 24 of file `ImageFormat2DBase.h`.

8.17.2.3 `using mappel::ImageFormat2DBase::ImagePixelT = double` `[inherited]`

Image pixel storage type

Definition at line 25 of file `ImageFormat2DBase.h`.

8.17.2.4 `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`.

8.17.2.5 `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`.

8.17.2.6 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` `[inherited]`

Data type for a single image size

Definition at line 29 of file `ImageFormat2DBase.h`.

8.17.2.7 `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.

8.17.2.8 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.17.2.9 `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.

8.17.2.10 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.17.2.11 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.17.2.12 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.17.2.13 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.17.2.14 `using mappel::Gauss2DsModel::StencilVecT = std::vector<Stencil>`

Definition at line 55 of file Gauss2DsModel.h.

8.17.3 Constructor & Destructor Documentation

8.17.3.1 `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()`.

8.17.3.2 `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`.

8.17.3.3 `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`.

8.17.4 Member Function Documentation

8.17.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const` `[inherited]`

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.17.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` `[inherited]`

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.17.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.17.4.4 `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`.

8.17.4.5 `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`.

8.17.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.17.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.17.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.17.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.17.4.10 `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()`.

8.17.4.11 `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()`.

8.17.4.12 `Gauss1DSumModelT mappel::Gauss2DsModel::debug_internal_sum_model_x () const` `[inline]`

Definition at line 104 of file Gauss2DsModel.h.

References `x_model`.

8.17.4.13 `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`.

8.17.4.14 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`, `[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

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

References `mappel::PointEmitterModel::prior`.

8.17.4.16 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline], [inherited]`

Definition at line 232 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.17.4.17 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline], [inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.17.4.18 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline], [inherited]`

Definition at line 106 of file `ImageFormat2DBase.h`.

8.17.4.19 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline], [inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.17.4.20 `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()`.

8.17.4.21 `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()`.

8.17.4.22 `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()`.

8.17.4.23 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::num_phases`.

8.17.4.24 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `mappel::MCMCAdaptorBase::sigma_scale`.

8.17.4.25 `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()`.

8.17.4.26 `double mappel::Gauss2DsModel::get_min_sigma (IdxT dim) const`

Definition at line 191 of file Gauss2DsModel.cpp.

References `DefaultPriorType`, `min_sigma`, and `prior_types`.

8.17.4.27 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`, `[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

8.17.4.28 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`, `[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.17.4.29 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().

8.17.4.30 StringVecT mappel::PointEmitterModel::get_param_names () const [inline],[inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and update_internal_1Dsum_↵
estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.17.4.35 const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const [inline],
[inherited]

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.17.4.36 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT *idx*) const
[*inherited*]

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

8.17.4.37 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & *stack*) const [inline],[*inherited*]

Definition at line 99 of file ImageFormat2DBase.h.

8.17.4.38 StatsT mappel::Gauss2DsModel::get_stats () const

Definition at line 341 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().

8.17.4.39 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[*inherited*]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and set_max_sigma_ratio().

8.17.4.40 bool mappel::PointEmitterModel::has_hyperparam (const std::string & *name*) const [inline],[*inherited*]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.17.4.41 Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & *im*) [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().

8.17.4.42 `Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init) [inline]`

Definition at line 231 of file Gauss2DsModel.h.

References `mappel::PointEmitterModel::DefaultSeperableInitEstimator`, and `initial_theta_estimate()`.

8.17.4.43 `Gauss2DsModel::Stencil mappel::Gauss2DsModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init, const std::string & estimator)`

Definition at line 385 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::PointEmitterModel::theta_in_bounds()`, `mappel::PointEmitterModel::ubound`, `x_model`, and `y_model`.

8.17.4.44 `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()`.

8.17.4.45 `CompositeDist mappel::Gauss2DsModel::make_default_prior_beta_position (const ImageSizeT & size, double max_sigma_ratio) [static]`

Definition at line 230 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_beta()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `make_default_prior()`.

8.17.4.46 `CompositeDist mappel::Gauss2DsModel::make_default_prior_normal_position (const ImageSizeT & size, double max_sigma_ratio) [static]`

Definition at line 243 of file Gauss2DsModel.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::PointEmitterModel::make_prior_component_intensity()`, `mappel::PointEmitterModel::make_prior_component_position_normal()`, `mappel::PointEmitterModel::make_prior_component_sigma()`, and `mappel::ImageFormat2DBase::size`.

Referenced by `make_default_prior()`.

8.17.4.47 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const [inline], [inherited]

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.17.4.48 ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT *n*) const [inline], [inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.17.4.49 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().

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

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DskyModel::initial_theta_estimate(), initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and pixel_hess_update().

8.17.4.51 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT *fill*) const [inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.17.4.52 MatT mappel::PointEmitterModel::make_param_mat () const [inline], [inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.17.4.53 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.17.4.54 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` [inline], [inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.17.4.55 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
[inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.17.4.56 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` [inline],
[inherited]

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.17.4.57 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` [inherited]

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.17.4.58 `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 255 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.

8.17.4.59 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_I, double kappa = default_intensity_kappa) [static],[inherited]

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_I.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and make_prior_normal_position().

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

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), make_default_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), and make_prior_beta_position().

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

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and make_prior_normal_position().

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), make_default_prior_beta_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), make_default_prior_normal_position(), mappel::Gauss1DsModel::make_prior_beta_position(), make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_normal_position(), and make_prior_normal_position().

8.17.4.63 **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 271 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.

8.17.4.64 `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\(\)](#).

8.17.4.65 `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=\(\)](#).

8.17.4.66 `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](#).

8.17.4.67 `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\(\)](#).

8.17.4.68 `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`.

8.17.4.69 `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`.

8.17.4.70 `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 358 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`.

8.17.4.71 `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`.

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

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.17.4.73 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta)`
`const [inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.17.4.74 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)`
`[inline],[inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.17.4.75 `void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double`
`step_scale = 1.0) [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`.

8.17.4.76 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.17.4.78 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1) [inherited]`

Definition at line 81 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::Point`
`EmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_`
`ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.17.4.79 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

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

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

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.17.4.81 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` `[inline]`,
`[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.17.4.82 `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()`.

8.17.4.83 `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`.

8.17.4.84 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1)` `[inherited]`

Definition at line 65 of file `MCMCAdaptor1D.cpp`.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.17.4.85 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` [inherited]

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.17.4.86 `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()`.

8.17.4.87 `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()`.

8.17.4.88 `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()`.

8.17.4.89 `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`.

8.17.4.90 `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`.

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

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.17.4.92 `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().

8.17.4.93 `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().

8.17.4.94 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` `[static]`, `[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.17.4.95 `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.

8.17.4.96 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss1DsModel::set_max_sigma(), and set_max_sigma_ratio().

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

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), initial_theta_estimate(), mappel::Gauss1DModel::make_stencil(), mappel::Gauss1DsModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DsxyModel::make_stencil(), make_stencil(), and mappel::PointEmitterModel::theta_stack_in_bounds().

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

Definition at line 297 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

8.17.4.99 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().

8.17.5 Member Data Documentation

8.17.5.1 const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static],[inherited]

Distance from the boundary to constrain in bound_theta and bounded_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), set_max_sigma↔_ratio(), and mappel::PointEmitterModel::set_ubound().

8.17.5.2 const double mappel::PointEmitterModel::default_alpha_sigma = 2 [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.17.5.3 const double mappel::PointEmitterModel::default_beta_pos = 3 [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.17.5.4 const double mappel::PointEmitterModel::default_intensity_kappa = 2 [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.17.5.5 const double mappel::PointEmitterModel::default_max_I = INFINITY [static],[inherited]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

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

8.17.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.17.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.17.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

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

Definition at line 59 of file Gauss2DsModel.h.

Referenced by `get_min_sigma()`.

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

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

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

8.17.5.11 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.17.5.12 `double mappel::MCMCAdaptor1D::eta_l=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.17.5.13 `double mappel::MCMCAdaptor2Ds::eta_sigma=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 26 of file `MCMCAdaptor2Ds.h`.

Referenced by `mappel::MCMCAdaptor2Ds::get_stats()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor2Ds::operator=()`, and `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`.

8.17.5.14 `double mappel::MCMCAdaptor1D::eta_x=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.17.5.15 `double mappel::MCMCAdaptor2D::eta_y=0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 27 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()`.

8.17.5.16 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale=0.05` `[static]`, `[inherited]`

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

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.17.5.17 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.17.5.18 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

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

8.17.5.19 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

8.17.5.20 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

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

8.17.5.21 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.17.5.23 VecT mappel::Gauss2DsModel::min_sigma [protected]

Gaussian PSF in pixels

Definition at line 118 of file Gauss2DsModel.h.

Referenced by compute_max_sigma_ratio(), debug_internal_sum_model_y(), mappel::Gauss2DsMAP::Gauss2DsMAP(), mappel::Gauss2DsMLE::Gauss2DsMLE(), Gauss2DsModel(), get_min_sigma(), initial_theta_estimate(), make_internal_1Dsum_estimator(), operator=(), set_min_sigma(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), and update_internal_1Dsum_estimators().

8.17.5.24 const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim =2 [static], [inherited]

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by mappel::ImageFormat2DBase::get_stats().

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

Definition at line 147 of file PointEmitterModel.h.

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

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

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::get_num_params(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), 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().

8.17.5.27 IdxT mappel::MCMCAdaptorBase::num_phases [protected], [inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_num_phases(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor2D::sample_mcmc_candidate(), mappel::MCMCAdaptor1Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor2Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor1D::sample_mcmc_candidate(), and mappel::MCMCAdaptorBase::set_mcmc_num_phases().

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

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), Gauss2DsModel(), mappel::PointEmitterModel::get_hyperparam_index(), mappel::PointEmitterModel::get_hyperparam_names(), mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_hyperparams(), mappel::Gauss1DsModel::get_max_sigma(), mappel::Gauss1DsModel::get_min_sigma(), mappel::PointEmitterModel::get_param_names(), mappel::PointEmitterModel::get_prior(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::has_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename_hyperparam(), mappel::PointEmitterModel::sample_prior(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_hyperparam_names(), mappel::PointEmitterModel::set_hyperparam_value(), mappel::PointEmitterModel::set_hyperparams(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::PointEmitterModel::set_param_names(), mappel::PointEmitterModel::set_prior(), and mappel::PointEmitterModel::set_ubound().

8.17.5.29 const StringVecT mappel::Gauss2DsModel::prior_types [static]

Initial value:

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

Definition at line 58 of file Gauss2DsModel.h.

Referenced by get_min_sigma().

8.17.5.30 double mappel::MCMCAdaptorBase::sigma_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_sigma_scale(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling(), and mappel::MCMCAdaptorBase::set_mcmc_sigma_scale().

8.17.5.31 **ImageSizeT** mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss_compute_estimate(), mappel::cgauss_compute_estimate_debug(), mappel::cgauss_←
_heuristic_compute_estimate(), mappel::ImageFormat2DBase::check_image_shape(), mappel::Gauss2DModel←
::Stencil::compute_derivatives(), mappel::Gauss2DsModel::Stencil::compute_derivatives(), mappel::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().

8.17.5.32 **ParamT** mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel←
::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta←
_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), initial_theta_estimate(), mappel::PointEmitterModel←
::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::Point←
EmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and
mappel::PointEmitterModel::theta_in_bounds().

8.17.5.33 **Gauss1DsumModelT** mappel::Gauss2DsModel::x_model [protected]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 119 of file Gauss2DsModel.h.

Referenced by debug_internal_sum_model_x(), initial_theta_estimate(), operator=(), set_max_sigma_ratio(), set_min←
_sigma(), set_size(), and update_internal_1Dsum_estimators().

8.17.5.34 Gauss1DSumModelT mappel::Gauss2DsModel::y_model [protected]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 120 of file Gauss2DsModel.h.

Referenced by debug_internal_sum_model_y(), Gauss2DsModel(), initial_theta_estimate(), operator=(), set_max_sigma_ratio(), set_min_sigma(), set_size(), and update_internal_1Dsum_estimators().

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

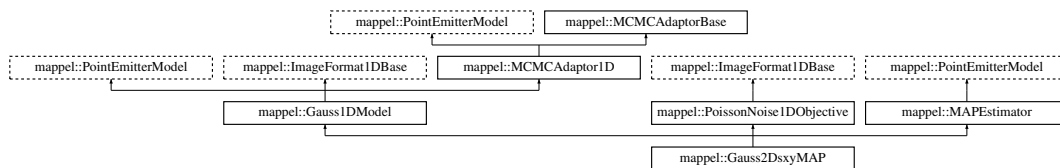
- [Gauss2DsModel.h](#)
- [Gauss2DsModel.cpp](#)

8.18 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)
- [ParamT sample_prior](#) ()
- 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)
- 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_↵, double sigma=[default_sigma_pos](#))
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_↵, double beta=[default_beta_pos](#))
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[default_mean_l](#)↵, double kappa=[default_intensity_kappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵sigma, double alpha=[default_alpha_sigma](#))
- static void [set_rng_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()
- static void [check_size](#) (const [ImageSizeT](#) &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [name](#)
- static const [StringVecT](#) [prior_types](#)
- static const std::string [DefaultPriorType](#) = "Normal"
- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) = 1
- static const [ImageCoordT](#) [global_min_size](#) = 3
- static const [ImageCoordT](#) [global_max_size](#) = 512
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5
- static const std::vector< std::string > [estimator_names](#)

Protected Member Functions

- void [set_mcmc_num_phases](#) (IdxT num_phases)

Protected Attributes

- double [psf_sigma](#)
- CompositeDist [prior](#)
- [IdxT](#) [num_params](#)
- [IdxT](#) [num_hyperparams](#)
- [ParamT](#) [lbound](#)
- [ParamT](#) [ubound](#)
- [ImageSizeT](#) [size](#)
- double [eta_x](#) =0
- double [eta_l](#) =0
- double [eta_bg](#) =0
- [IdxT](#) [num_phases](#)
- double [sigma_scale](#)

8.18.1 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.

8.18.2 Member Typedef Documentation

8.18.2.1 `using mappel::ImageFormat1DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

8.18.2.2 `using mappel::ImageFormat1DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

8.18.2.3 `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.

8.18.2.4 `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.

8.18.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
[inherited]

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.18.2.6 `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.

8.18.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
[inherited]

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.18.2.8 `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.

8.18.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
[inherited]

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.18.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` [inherited]

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.18.2.11 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.

8.18.2.12 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.

8.18.2.13 using mappel::PointEmitterModel::ParamT = arma::vec [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.18.2.14 using mappel::PointEmitterModel::ParamVecT = arma::mat [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.18.2.15 using mappel::Gauss1DModel::StencilVecT = std::vector<Stencil> [inherited]

Definition at line 49 of file Gauss1DModel.h.

8.18.3 Constructor & Destructor Documentation

8.18.3.1 mappel::Gauss2DsxyMAP::Gauss2DsxyMAP (const ImageSizeT & size, const VecT & min_sigma, const VecT & max_sigma)

8.18.3.2 mappel::Gauss2DsxyMAP::Gauss2DsxyMAP (const ImageSizeT & size, const VecT & min_sigma, const VecT & max_sigma, CompositeDist && prior)

8.18.4 Member Function Documentation

8.18.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

8.18.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.18.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.18.4.4 `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`.

8.18.4.5 `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`.

8.18.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.18.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.18.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` `[inherited]`

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.18.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` `[inherited]`

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.18.4.10 `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()`.

8.18.4.11 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`, `[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 256 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.18.4.13 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`, `[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, and `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.18.4.14 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.18.4.15 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack,`
`ImageCoordT n) const` `[inline]`, `[inherited]`

Definition at line 108 of file `ImageFormat1DBase.h`.

8.18.4.16 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.18.4.17 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file `MCMCAdaptorBase.cpp`.

References `mappel::MCMCAdaptorBase::num_phases`.

8.18.4.18 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file `MCMCAdaptorBase.cpp`.

References `mappel::MCMCAdaptorBase::sigma_scale`.

8.18.4.19 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`, `[inherited]`

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

References `mappel::PointEmitterModel::num_hyperparams`.

8.18.4.20 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`, `[inherited]`

Definition at line 160 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.18.4.21 `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().

8.18.4.22 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline]`,`[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

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

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.18.4.25 `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().

8.18.4.26 `double mappel::Gauss1DModel::get_psf_sigma (IdxT idx) const` `[inherited]`

Definition at line 131 of file Gauss1DModel.cpp.

References mappel::Gauss1DModel::psf_sigma.

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

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.18.4.29 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().

8.18.4.30 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT idx) const [inherited]

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

8.18.4.31 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]

Definition at line 101 of file ImageFormat1DBase.h.

8.18.4.32 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().

8.18.4.33 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.18.4.34 bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.18.4.35 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.

8.18.4.36 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.

8.18.4.37 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().

8.18.4.38 CompositeDist mappel::Gauss1DModel::make_default_prior_beta_position (IdxT *size*) [static], [inherited]

Definition at line 80 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default_pixel_mean_bg, mappel::PointEmitterModel::make_prior_component_intensity(), and mappel::PointEmitterModel::make_prior_component_position_beta().

Referenced by mappel::Gauss1DModel::make_default_prior().

8.18.4.39 CompositeDist mappel::Gauss1DModel::make_default_prior_normal_position (IdxT *size*) [static], [inherited]

Definition at line 90 of file Gauss1DModel.cpp.

References mappel::PointEmitterModel::default_pixel_mean_bg, mappel::PointEmitterModel::make_prior_component_intensity(), and mappel::PointEmitterModel::make_prior_component_position_normal().

Referenced by mappel::Gauss1DModel::make_default_prior().

8.18.4.40 ImageFormat1DBase::ImageT ImageFormat1DBase::make_image () const [inline], [inherited]

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

8.18.4.41 ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack (ImageCoordT *n*) const
`[inline],[inherited]`

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

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

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.18.4.43 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT *fill*) const
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.18.4.44 MatT mappel::PointEmitterModel::make_param_mat () const `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.18.4.45 template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT *fill*) const `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.18.4.46 CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT *n*) const `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.18.4.47 template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT *n*, FillT *fill*) const
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.18.4.48 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline]`,
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

8.18.4.49 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.18.4.50 `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()`.

8.18.4.51 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.18.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` `[static]`, `[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

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

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

Definition at line 85 of file PointEmitterModel.cpp.

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

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

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.18.4.55 `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()`.

8.18.4.56 `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\(\)](#).

```
8.18.4.57 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\(\)](#).

```
8.18.4.58 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](#).

```
8.18.4.59 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](#).

```
8.18.4.60 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](#).

```
8.18.4.61 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](#).

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

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

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

8.18.4.63 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const`
`[inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.18.4.64 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)`
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.18.4.65 `void mappel::MCMCAdaptor1D::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)`
`[inherited]`

Definition at line 108 of file `MCMCAdaptor1D.cpp`.

References `mappel::MCMCAdaptor1D::eta_bg`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::MCMCAdaptor1D::eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

8.18.4.66 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.18.4.68 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.18.4.69 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

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

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

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.18.4.71 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` [inline], [inherited]

Definition at line 240 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.18.4.72 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` [inline], [inherited]

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

8.18.4.73 `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.

8.18.4.74 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_l = -1)` `[inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_l`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.18.4.75 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.18.4.76 `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()`.

8.18.4.77 `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`.

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

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

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

Definition at line 158 of file PointEmitterModel.cpp.

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

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

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

Definition at line 149 of file PointEmitterModel.cpp.

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

8.18.4.81 `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()`.

8.18.4.82 `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()`.

8.18.4.83 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` [static],[inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.18.4.84 `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()`.

8.18.4.85 `void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & sz)` [inline],[inherited]

Definition at line 75 of file ImageFormat1DBase.h.

References `mappel::ImageFormat1DBase::set_size()`.

8.18.4.86 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

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

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

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

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.18.4.88 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const` `[inherited]`

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.18.5 Member Data Documentation

8.18.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static]`, `[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.18.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static]`, `[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.18.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static]`, `[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.18.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static]`, `[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.18.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::make_prior_component_intensity().

8.18.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.18.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.18.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.18.5.9 `const std::string mappel::Gauss1DModel::DefaultPriorType = "Normal"` `[static], [inherited]`

Definition at line 53 of file Gauss1DModel.h.

Referenced by mappel::Gauss1DModel::operator=().

8.18.5.10 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), and mappel::Gauss2DsModel::initial_theta_estimate().

8.18.5.11 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names` `[static], [inherited]`

Definition at line 24 of file PoissonNoise1DObjective.h.

8.18.5.12 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.18.5.13 `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.18.5.14 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.18.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.18.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.18.5.17 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.18.5.18 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_max_size = 512` `[static], [inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 40 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

8.18.5.19 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.18.5.20 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::global_min_size = 3` `[static], [inherited]`

Minimum size along any dimension of the image.

Definition at line 39 of file `ImageFormat1DBase.h`.

Referenced by `mappel::ImageFormat1DBase::check_size()`.

8.18.5.21 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.18.5.22 `const std::string mappel::Gauss2DsxyMAP::name` `[static]`

Definition at line 30 of file `Gauss2DsxyMAP.h`.

8.18.5.23 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static],[inherited]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::get_stats()`.

8.18.5.24 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected],[inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

8.18.5.25 `IdxT mappel::PointEmitterModel::num_params` `[protected],[inherited]`

Definition at line 146 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.18.5.26 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected],[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

8.18.5.27 `CompositeDist mappel::PointEmitterModel::prior` `[protected],[inherited]`

Definition at line 145 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.18.5.28 `const StringVect mappel::Gauss1DModel::prior_types` `[static], [inherited]`

Initial value:

```
= { "Beta",  
                                     "Normal"  
}
```

Definition at line 52 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::operator=()`.

8.18.5.29 `double mappel::Gauss1DModel::psf_sigma` `[protected], [inherited]`

Standard deviation of the fixed-sigma 1D Gaussian PSF in pixels

Definition at line 90 of file Gauss1DModel.h.

Referenced by `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::get_psf_sigma()`, `mappel::Gauss1DModel::operator=()`, `mappel::Gauss1DModel::pixel_grad2()`, `mappel::Gauss1DModel::pixel_hess()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

8.18.5.30 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected], [inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.18.5.31 `ImageSizeT mappel::ImageFormat1DBase::size` `[protected], [inherited]`

Number of pixels in X dimension for 1D image

Definition at line 65 of file ImageFormat1DBase.h.

Referenced by `mappel::ImageFormat1DBase::check_image_shape()`, `mappel::Gauss1DsModel::Stencil::compute_derivatives()`, `mappel::Gauss1DModel::Stencil::compute_derivatives()`, `mappel::ImageFormat1DBase::get_num_pixels()`, `mappel::ImageFormat1DBase::get_size()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::ImageFormat1DBase::make_image()`, `mappel::ImageFormat1DBase::make_image_stack()`, `mappel::ImageFormat1DBase::set_size()`, `mappel::Gauss1DsModel::Stencil::Stencil()`, and `mappel::Gauss1DModel::Stencil::Stencil()`.

8.18.5.32 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

The documentation for this class was generated from the following file:

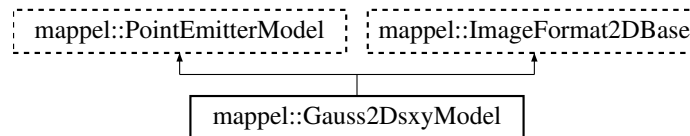
- [Gauss2DsxyMAP.h](#)

8.19 mappel::Gauss2DsxyModel 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/Gauss2DsxyModel.h>
```

Inheritance diagram for mappel::Gauss2DsxyModel:



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)
- `ParamT` `sample_prior` ()
- void `set_bounds` (const `ParamT` &lbound, const `ParamT` &ubound)
- void `set_lbound` (const `ParamT` &lbound)
- void `set_ubound` (const `ParamT` &ubound)
- const `ParamT` & `get_lbound` () const
- const `ParamT` & `get_ubound` () const
- bool `theta_in_bounds` (const `ParamT` &theta) const
- void `bound_theta` (`ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `bounded_theta` (const `ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `reflected_theta` (const `ParamT` &theta) const
- `BoolVecT` `theta_stack_in_bounds` (const `ParamVecT` &theta) const
- `ParamVecT` `bounded_theta_stack` (const `ParamVecT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamVecT` `reflected_theta_stack` (const `ParamVecT` &theta) const
- `ImageT` `make_image` () const
- `ImageStackT` `make_image_stack` (`ImageCoordT` n) const
- `ImageCoordT` `get_size_image_stack` (const `ImageStackT` &stack) const
- `ImageT` `get_image_from_stack` (const `ImageStackT` &stack, `ImageCoordT` n) const
- template<class ImT >
 void `set_image_in_stack` (`ImageStackT` &stack, `ImageCoordT` n, const ImT &im) const
- const `ImageSizeT` & `get_size` () const
- `ImageCoordT` `get_size` (`IdxT` idx) const
- `ImageCoordT` `get_num_pixels` () const
- void `check_image_shape` (const `ImageT` &im) const
 Check the shape of a single images is correct for model size.
- void `check_image_shape` (const `ImageStackT` &ims) const
 Check the shape of a stack of images is correct for model size.

Static Public Member Functions

- static `CompositeDist` `make_default_prior` (const `ImageSizeT` &size, double max_sigma_ratio)
- static `CompositeDist` `make_prior_beta_position` (const `ImageSizeT` &size, double beta_xpos, double beta_ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_←sigma)
- static `CompositeDist` `make_prior_normal_position` (const `ImageSizeT` &size, double sigma_xpos, double sigma_←ypos, double mean_l, double kappa_l, double mean_bg, double kappa_bg, double max_sigma_ratio, double alpha_sigma)
- static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` size, double pos_←sigma=default_sigma_pos)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos_←beta=default_beta_pos)

- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[default_mean_↵](#)
[I](#), double kappa=[default_intensity_kappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double [min_sigma](#), double max_[↵](#)
[sigma](#), double alpha=[default_alpha_sigma](#))
- static void [set_rng_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()
- static void [check_size](#) (const [ImageSizeT](#) &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_I](#) = 300
- static const double [default_max_I](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) [num_dim](#) =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](#)

8.19.1 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`.

8.19.2 Member Typedef Documentation

8.19.2.1 `using mappel::ImageFormat2DBase::ImageCoordT = uint32_t` [inherited]

Image size coordinate storage type

Definition at line 24 of file `ImageFormat2DBase.h`.

8.19.2.2 `using mappel::ImageFormat2DBase::ImagePixelT = double` [inherited]

Image pixel storage type

Definition at line 25 of file `ImageFormat2DBase.h`.

8.19.2.3 `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`.

8.19.2.4 `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`.

8.19.2.5 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>` [inherited]

Data type for a single image size

Definition at line 29 of file `ImageFormat2DBase.h`.

8.19.2.6 `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.

8.19.2.7 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.19.2.8 `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.

8.19.2.9 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.19.2.10 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.19.2.11 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.19.2.12 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.19.2.13 `using mappel::Gauss2DsxyModel::StencilVecT = std::vector<Stencil>`

Definition at line 59 of file Gauss2DsxyModel.h.

8.19.3 Constructor & Destructor Documentation

8.19.3.1 `mappel::Gauss2DskyModel::Gauss2DskyModel (const ImageSizeT & size, const VecT & min_sigma, const VecT & max_sigma)`

8.19.4 Member Function Documentation

8.19.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.19.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

8.19.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.19.4.4 `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`.

8.19.4.5 `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`.

8.19.4.6 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.19.4.7 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.19.4.8 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.19.4.9 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.19.4.10 `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().

8.19.4.11 `static double mappel::Gauss2DsxModel::compute_max_sigma_ratio (const VecT & min_sigma, const VecT & max_sigma) [static],[protected]`

8.19.4.12 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const [inline],[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.19.4.13 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const [inline],[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.19.4.14 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const [inline],[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.19.4.15 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const [inline],[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.19.4.16 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const [inline],[inherited]`

Definition at line 106 of file ImageFormat2DBase.h.

8.19.4.17 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const [inline],[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.19.4.18 VecT mappel::Gauss2DskyModel::get_max_sigma () const [inline]

Definition at line 127 of file Gauss2DskyModel.h.

References `get_max_sigma_ratio()`, and `get_min_sigma()`.

8.19.4.19 double mappel::Gauss2DskyModel::get_max_sigma (IdxT dim) const [inline]

Definition at line 131 of file Gauss2DskyModel.h.

References `get_max_sigma_ratio()`, and `get_min_sigma()`.

8.19.4.20 double mappel::Gauss2DskyModel::get_max_sigma_ratio () const [inline]

Definition at line 135 of file Gauss2DskyModel.h.

References `mappel::PointEmitterModel::get_ubound()`.

Referenced by `get_max_sigma()`.

8.19.4.21 VecT mappel::Gauss2DskyModel::get_min_sigma () const [inline]

Definition at line 122 of file Gauss2DskyModel.h.

References `min_sigma`.

Referenced by `get_max_sigma()`.

8.19.4.22 double mappel::Gauss2DskyModel::get_min_sigma (IdxT dim) const**8.19.4.23 IdxT mappel::PointEmitterModel::get_num_hyperparams () const [inline],[inherited]**

Definition at line 208 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_hyperparams`.

8.19.4.24 IdxT mappel::PointEmitterModel::get_num_params () const [inline],[inherited]

Definition at line 160 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.19.4.25 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()`.

8.19.4.26 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.19.4.27 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline],[inherited]`

Definition at line 200 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

8.19.4.28 `const CompositeDist & mappel::PointEmitterModel::get_prior () const` `[inline],[inherited]`

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.19.4.29 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static],[inherited]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.19.4.30 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()` `[static],[inherited]`

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.19.4.31 `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const` `[inline],[inherited]`

Definition at line 74 of file `ImageFormat2DBase.h`.

References `mappel::ImageFormat2DBase::size`.

8.19.4.32 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT idx) const` `[inherited]`

Definition at line 41 of file `ImageFormat2DBase.cpp`.

References `mappel::ImageFormat2DBase::size`.

8.19.4.33 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & stack) const [inline],[inherited]

Definition at line 99 of file ImageFormat2DBase.h.

8.19.4.34 StatsT mappel::Gauss2DsxyModel::get_stats () const

8.19.4.35 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.19.4.36 bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.19.4.37 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().

8.19.4.38 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().

8.19.4.39 Stencil mappel::Gauss2DsxyModel::initial_theta_estimate (const ImageT & im, const ParamT & theta_init, const std::string & estimator)

8.19.4.40 static CompositeDist mappel::Gauss2DsxyModel::make_default_prior (const ImageSizeT & size, double max_sigma_ratio) [static]

8.19.4.41 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const [inline],[inherited]

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.19.4.42 ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT *n*) const [inline],[inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.19.4.43 PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const [inline],[inherited]

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.19.4.44 template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT *fill*) const [inherited]

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.19.4.45 MatT mappel::PointEmitterModel::make_param_mat () const [inline],[inherited]

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.19.4.46 template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT *fill*) const [inherited]

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.19.4.47 CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT *n*) const [inline],[inherited]

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.19.4.48 template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT *n*, FillT *fill*) const [inherited]

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.19.4.49 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline]`,
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

8.19.4.50 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.19.4.51 `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]`

8.19.4.52 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.19.4.53 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` `[static]`, `[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

8.19.4.54 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal (IdxT size, double pos_sigma = default_sigma_pos) [static],[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.19.4.55 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma (double min_sigma, double max_sigma, double alpha = default_alpha_sigma) [static],[inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.19.4.56 `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]`

8.19.4.57 `Gauss2DsxyModel::Stencil mappel::Gauss2DsxyModel::make_stencil (const ParamT & theta, bool compute_derivatives = true) const [inline]`

Make a new `Model::Stencil` object at `theta`.

Stencils store all of the important calculations necessary for evaluating the log-likelihood and its derivatives at a particular `theta` (parameter) value.

This allows re-use of the most expensive computations. Stencils can be easily passed around by reference, and most functions in the `mappel::methods` namespace accept a const `Stencil` reference in place of the model parameter.

Throws `mappel::ModelBoundsError` if not `model.theta_in_bounds(theta)`.

If derivatives will not be computed with this stencil set `compute_derivatives=false`

Parameters

<i>theta</i>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

Returns

A new `Stencil` object ready to compute with

Definition at line 157 of file Gauss2DsxModel.h.

References mappel::Gauss2DsxModel::Stencil::Stencil(), and mappel::PointEmitterModel::theta_in_bounds().

8.19.4.58 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::I(), mappel::Gauss2DsxModel::Stencil::X, and mappel::Gauss2DsxModel::Stencil::Y.

8.19.4.59 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::I(), mappel::Gauss2DsxModel::Stencil::sigmaX(), mappel::Gauss2DsxModel::Stencil::sigmaY(), mappel::Gauss2DsxModel::Stencil::X, and mappel::Gauss2DsxModel::Stencil::Y.

8.19.4.60 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::I(), mappel::Gauss2DsxModel::Stencil::sigmaX(), mappel::Gauss2DsxModel::Stencil::sigmaY(), mappel::Gauss2DsxModel::Stencil::X, and mappel::Gauss2DsxModel::Stencil::Y.

8.19.4.61 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

8.19.4.62 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::I(), mappel::Gauss2DsxModel::Stencil::X, and mappel::Gauss2DsxModel::Stencil::Y.

8.19.4.63 PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & *theta*) const [inherited]

Definition at line 276 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::reflected_theta_stack().

8.19.4.64 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta)`
`const [inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.19.4.65 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)`
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.19.4.66 `void mappel::Gauss2DsxxyModel::sample_mcmc_candidate (int sample_index, ParamT & candidate_theta, double scale = 1.0)`

8.19.4.67 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.19.4.68 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () [inline], [inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.19.4.69 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.19.4.70 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc) [inline],`
`[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.19.4.71 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value) [inline], [inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.19.4.72 `void mappel::Gauss2DsxyModel::set_hyperparams (const VecT & hyperparams)`

8.19.4.73 `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.

8.19.4.74 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set_min_sigma().

8.19.4.75 `void mappel::Gauss2DsxyModel::set_max_sigma (const VecT & max_sigma)`

8.19.4.76 `void mappel::Gauss2DsxyModel::set_max_sigma_ratio (double max_sigma_ratio)`

8.19.4.77 `void mappel::Gauss2DsxyModel::set_min_sigma (const VecT & min_sigma)`

8.19.4.78 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc) [inline], [inherited]`

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.19.4.79 `void mappel::Gauss2DsxyModel::set_prior (CompositeDist && prior_)`

8.19.4.80 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_) [inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_hyperparams, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

8.19.4.81 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static], [inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.19.4.82 `void mappel::Gauss2Dsxymodel::set_size (const ImageSizeT & size_)`

8.19.4.83 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

8.19.4.84 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const` [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.19.4.85 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const` [inherited]

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.19.4.86 `void mappel::Gauss2Dsxymodel::update_internal_1D_estimators ()` [protected]

8.19.5 Member Data Documentation

8.19.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static], [inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.19.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static], [inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.19.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.19.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.19.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

8.19.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.19.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.19.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.19.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],`
`[inherited]`

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `initial_theta_estimate()`, and `mappel::Gauss2Ds←
Model::initial_theta_estimate()`.

8.19.5.10 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2 [static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.19.5.11 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_max_size =512 [static],`
`[inherited]`

Maximum size along any dimension of the image. This is insanely big to catch obvious errors

Definition at line 39 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

8.19.5.12 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1 [static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.19.5.13 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::global_min_size =3 [static],`
`[inherited]`

Minimum size along any dimension of the image.

Definition at line 38 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::check_size()`.

8.19.5.14 `ParamT mappel::PointEmitterModel::lbound` `[protected]`, `[inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.19.5.15 `double mappel::Gauss2DsxyModel::mcmc_candidate_eta_sigma` `[protected]`

The standard deviation for the normally distributed perturbation to `theta_sigma` in the random walk MCMC sampling

Definition at line 108 of file Gauss2DsxyModel.h.

8.19.5.16 `double mappel::Gauss2DsxyModel::mcmc_candidate_eta_y` `[protected]`

Std-dev for the normal perturbations to `theta_y` under MCMC sampling

Definition at line 107 of file Gauss2DsxyModel.h.

8.19.5.17 `VecT mappel::Gauss2DsxyModel::min_sigma` `[protected]`

Gaussian PSF in pixels

Definition at line 113 of file Gauss2DsxyModel.h.

Referenced by `get_min_sigma()`.

8.19.5.18 `const ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::num_dim = 2` `[static]`, `[inherited]`

Number of image dimensions.

Definition at line 37 of file ImageFormat2DBase.h.

Referenced by `mappel::ImageFormat2DBase::get_stats()`.

8.19.5.19 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`, `[inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

8.19.5.20 IdxT mappel::PointEmitterModel::num_params [protected],[inherited]

Definition at line 146 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::get_num_params(), mappel::PointEmitterModel::get_stats(), mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::make_param(), mappel::PointEmitterModel::make_param_mat(), mappel::PointEmitterModel::make_param_mat_stack(), mappel::PointEmitterModel::make_param_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.19.5.21 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get_hyperparam_index(), mappel::PointEmitterModel::get_hyperparam_names(), mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_hyperparams(), mappel::Gauss1DsModel::get_max_sigma(), mappel::Gauss1DsModel::get_min_sigma(), mappel::PointEmitterModel::get_param_names(), mappel::PointEmitterModel::get_prior(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::has_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename_hyperparam(), mappel::PointEmitterModel::sample_prior(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_hyperparam_names(), mappel::PointEmitterModel::set_hyperparam_value(), mappel::PointEmitterModel::set_hyperparams(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::PointEmitterModel::set_param_names(), mappel::PointEmitterModel::set_prior(), and mappel::PointEmitterModel::set_ubound().

8.19.5.22 ImageSizeT mappel::ImageFormat2DBase::size [protected],[inherited]

Number of pixels in X dimension for 1D image

Definition at line 67 of file ImageFormat2DBase.h.

Referenced by mappel::cgauss_compute_estimate(), mappel::cgauss_compute_estimate_debug(), mappel::cgauss_heuristic_compute_estimate(), mappel::ImageFormat2DBase::check_image_shape(), mappel::Gauss2DModel::Stencil::compute_derivatives(), mappel::Gauss2DsModel::Stencil::compute_derivatives(), mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::methods::expected_information(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsMAP::Gauss2DsMAP(), mappel::Gauss2DsMLE::Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::ImageFormat2DBase::get_num_pixels(), mappel::ImageFormat2DBase::get_size(), mappel::ImageFormat2DBase::get_stats(), mappel::methods::likelihood::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood::debug::grad_components(), mappel::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian_components(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh_components(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::ImageFormat2DBase::make_image(), mappel::ImageFormat2DBase::make_image_stack(),

mappel::Gauss2DModel::make_internal_1Dsum_estimator(), mappel::Gauss2DsModel::make_internal_1Dsum_estimator(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss2DsModel::make_prior_normal_position(), mappel::methods::model_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods::likelihood::rllh(), mappel::methods::likelihood::debug::rllh_components(), mappel::ImageFormat2DBase::set_size(), mappel::Gauss2DModel::set_size(), mappel::Gauss2DsModel::set_size(), mappel::methods::simulate_image(), mappel::methods::simulate_image_from_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2DsModel::Stencil::Stencil(), mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

8.19.5.23 ParamT mappel::PointEmitterModel::ubound [protected], [inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

8.19.5.24 Gauss1DsMAP mappel::Gauss2Dsxymodel::x_model [protected]

X-model fits 2D images X-axis (column sum). Using variable sigma 1D model.

Definition at line 114 of file Gauss2Dsxymodel.h.

8.19.5.25 Gauss1DsMAP mappel::Gauss2Dsxymodel::y_model [protected]

Y-model fits 2D images Y-axis (row sum). Using variable sigma 1D model.

Definition at line 115 of file Gauss2Dsxymodel.h.

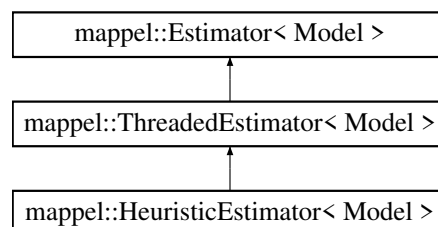
The documentation for this class was generated from the following file:

- [Gauss2Dsxymodel.h](#)

8.20 mappel::HeuristicEstimator< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::HeuristicEstimator< Model >:



Public Types

- enum `ExitCode` : `IdxT` {
`ExitCode::Unassigned` = 8, `ExitCode::MaxIter` = 7, `ExitCode::MaxBacktracks` = 6, `ExitCode::TrustRegionRadius` = 5,
`ExitCode::GradRatio` = 4, `ExitCode::FunctionChange` = 3, `ExitCode::StepSize` = 2, `ExitCode::Success` = 1,
`ExitCode::Error` = 0 }

Public Member Functions

- `HeuristicEstimator` (`Model` &`model`)
- `StatsT` `get_stats` ()
- `StatsT` `get_debug_stats` ()
- `std::string` `name` () const
- void `estimate_max_stack` (const `ModelDataStackT`< `Model` > &`im`, const `ParamVecT`< `Model` > &`theta_init`, `ParamVecT`< `Model` > &`theta`, `VecT` &`rllh`, `CubeT` &`obsI`)
- void `estimate_max_stack` (const `ModelDataStackT`< `Model` > &`im_stack`, `ParamVecT`< `Model` > &`theta_est`, `VecT` &`rllh_stack`, `CubeT` &`obsI_stack`)
- void `estimate_profile_stack` (const `ModelDataT`< `Model` > &`data`, const `IdxVecT` &`fixed_parameters`, const `MatT` &`values`, const `ParamVecT`< `Model` > &`theta_init`, `VecT` &`profile_likelihood`, `ParamVecT`< `Model` > &`profile_parameters`)
- void `clear_stats` ()
- `Model` & `get_model` ()
- void `set_model` (`Model` &`new_model`)
- `StencilT`< `Model` > `estimate_max` (const `ModelDataT`< `Model` > &`im`)
- `StencilT`< `Model` > `estimate_max` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`)
- `StencilT`< `Model` > `estimate_max` (const `ModelDataT`< `Model` > &`im`, double &`rllh`)
- `StencilT`< `Model` > `estimate_max` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, double &`rllh`)
- void `estimate_max` (const `ModelDataT`< `Model` > &`im`, `ParamT`< `Model` > &`theta`, double &`rllh`, `MatT` &`obsI`)
- void `estimate_max` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, `ParamT`< `Model` > &`theta`, double &`rllh`, `MatT` &`obsI`)
- void `estimate_max_debug` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, `ParamT`< `Model` > &`theta_est`, double &`rllh`, `MatT` &`obsI`, `MatT` &`sequence`, `VecT` &`sequence_rllh`)

Static Public Attributes

- static constexpr int `NumExitCodes` = 9

Protected Member Functions

- void `record_exit_code` (`ExitCode` `code`)
- virtual void `compute_estimate` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, `ParamT`< `Model` > &`theta_est`, double &`rllh`, `MatT` &`obsI`)
Default base class implementation computes rllh and obsI separately from stencil This should be overridden by Estimator subclasses that already have access to this information.
- virtual `StencilT`< `Model` > `compute_estimate_debug` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, `ParamVecT`< `Model` > &`sequence`, `VecT` &`sequence_rllh`)
- virtual void `compute_profile_estimate` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, const `IdxVecT` &`fixed_parameters`, `ParamT`< `Model` > &`theta_est`, double &`rllh`)
- void `record_walltime` (`ClockT::time_point` `start_walltime`, int `nimages`)

Protected Attributes

- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.
- [IdxVecT](#) [exit_counts](#)

8.20.1 Detailed Description

```
template<class Model>
class mappel::HeuristicEstimator< Model >
```

Definition at line 150 of file estimator.h.

8.20.2 Member Enumeration Documentation

8.20.2.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT` `[strong], [inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.20.3 Constructor & Destructor Documentation

8.20.3.1 `template<class Model > mappel::HeuristicEstimator< Model >::HeuristicEstimator (Model & model)`
`[inline]`

Definition at line 152 of file estimator.h.

References `mappel::Estimator< Model >::get_debug_stats()`, and `mappel::Estimator< Model >::get_stats()`.

8.20.4 Member Function Documentation

8.20.4.1 `template<class Model> void mappel::ThreadedEstimator< Model>::clear_stats () [virtual], [inherited]`

Reimplemented from [mappel::Estimator< Model>](#).

Reimplemented in [mappel::IterativeMaximizer< Model>](#).

Definition at line 378 of file `estimator_impl.h`.

References [mappel::Estimator< Model>::clear_stats\(\)](#), [mappel::ThreadedEstimator< Model>::mtx](#), and [mappel::ThreadedEstimator< Model>::num_threads](#).

Referenced by [mappel::IterativeMaximizer< Model>::clear_stats\(\)](#).

8.20.4.2 `template<class Model> void mappel::Estimator< Model>::compute_estimate (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta_est, double & rllh, MatT & obsl) [protected], [virtual], [inherited]`

Default base class implementation computes rllh and obsl separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file `estimator_impl.h`.

References [mappel::Estimator< Model>::compute_estimate\(\)](#), [mappel::Estimator< Model>::model](#), and [mappel::methods::observed_information\(\)](#).

8.20.4.3 `template<class Model> StencilT< Model> mappel::Estimator< Model>::compute_estimate_debug (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamVecT< Model> & sequence, VecT & sequence_rllh) [protected], [virtual], [inherited]`

Estimators that produce a sequence of results (e.g. [IterativeEstimators](#)) can override this dummy debug implementation.

Reimplemented in [mappel::IterativeMaximizer< Model>](#), [mappel::SimulatedAnnealingMaximizer< Model>](#), and [mappel::CGaussMLE< Model>](#).

Definition at line 242 of file `estimator_impl.h`.

References [mappel::Estimator< Model>::compute_estimate\(\)](#), [mappel::Estimator< Model>::Error](#), [mappel::Estimator< Model>::model](#), [mappel::print_image\(\)](#), and [mappel::Estimator< Model>::record_exit_code\(\)](#).

Referenced by [mappel::Estimator< Model>::estimate_max_debug\(\)](#), [mappel::CGaussMLE< Model>::name\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model>::name\(\)](#), and [mappel::Estimator< Model>::~~Estimator\(\)](#).

8.20.4.4 `template<class Model> void mappel::Estimator< Model>::compute_profile_estimate (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, const IdxVecT & fixed_parameters, ParamT< Model> & theta_est, double & rllh) [protected], [virtual], [inherited]`

Reimplemented in [mappel::IterativeMaximizer< Model>](#).

Definition at line 192 of file `estimator_impl.h`.

References [mappel::Estimator< Model>::model](#).

Referenced by [mappel::ThreadedEstimator< Model>::estimate_profile_stack\(\)](#), and [mappel::Estimator< Model>::~~Estimator\(\)](#).

8.20.4.5 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im)` [inherited]

Definition at line 64 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`.

Referenced by `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::~Estimator()`.

8.20.4.6 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init)` [inherited]

Definition at line 82 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::methods::objective::rllh()`.

8.20.4.7 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh)` [inherited]

Definition at line 73 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.20.4.8 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh)` [inherited]

Definition at line 90 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.20.4.9 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl)` [inherited]

Definition at line 119 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.20.4.10 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta, double & rllh, MatT & obsl)` [inherited]

Definition at line 128 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

8.20.4.11 `template<class Model> void mappel::Estimator< Model>::estimate_max_debug (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate_debug()`, `mappel::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::rllh()`.

Referenced by `mappel::Estimator< Model>::~~Estimator()`.

8.20.4.12 `template<class Model> void mappel::Estimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im_stack, ParamVecT< Model> & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack) [inherited]`

Definition at line 170 of file estimator_impl.h.

References `mappel::Estimator< Model>::estimate_max_stack()`, and `mappel::Estimator< Model>::model`.

8.20.4.13 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im, const ParamVecT< Model> & theta_init, ParamVecT< Model> & theta, VecT & rllh, CubeT & obsl) [virtual], [inherited]`

Implements [mappel::Estimator< Model>](#).

Definition at line 285 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.20.4.14 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_profile_stack (const ModelDataT< Model> & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model> & theta_init, VecT & profile_likelihood, ParamVecT< Model> & profile_parameters) [virtual], [inherited]`

Implements [mappel::Estimator< Model>](#).

Definition at line 323 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::record_walltime()`.

8.20.4.15 `template<class Model> StatsT mappel::HeuristicEstimator< Model>::get_debug_stats () [virtual]`

Reimplemented from [mappel::ThreadedEstimator< Model>](#).

Definition at line 414 of file estimator_impl.h.

References `mappel::cgauss_heuristic_compute_estimate()`, `mappel::ThreadedEstimator< Model>::get_stats()`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::methods::objective::rllh()`.

8.20.4.16 `template<class Model > Model & mappel::Estimator< Model >::get_model ()` [inherited]

Definition at line 50 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.20.4.17 `template<class Model > StatsT mappel::HeuristicEstimator< Model >::get_stats ()` [virtual]

Reimplemented from `mappel::ThreadedEstimator< Model >`.

Definition at line 404 of file estimator_impl.h.

References `mappel::ThreadedEstimator< Model >::get_stats()`.

8.20.4.18 `template<class Model > std::string mappel::HeuristicEstimator< Model >::name () const` [inline], [virtual]

Implements `mappel::Estimator< Model >`.

Definition at line 156 of file estimator.h.

References `mappel::Estimator< Model >::compute_estimate()`, and `mappel::methods::objective::rllh()`.

8.20.4.19 `template<class Model > void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code)` [protected], [virtual], [inherited]

Implements `mappel::Estimator< Model >`.

Definition at line 386 of file estimator_impl.h.

References `mappel::Estimator< Model >::exit_counts`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::mtx`, and `mappel::methods::objective::rllh()`.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::maximize()`.

8.20.4.20 `template<class Model > void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages)` [protected], [inherited]

Definition at line 266 of file estimator_impl.h.

References `mappel::Estimator< Model >::num_estimations`, and `mappel::Estimator< Model >::total_walltime`.

Referenced by `mappel::Estimator< Model >::estimate_max()`, `mappel::Estimator< Model >::estimate_max_debug()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, and `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`.

8.20.4.21 `template<class Model > void mappel::Estimator< Model >::set_model (Model & new_model)` [inherited]

Definition at line 54 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::~~Estimator().

8.20.5 Member Data Documentation

8.20.5.1 `template<class Model > IdxVecT mappel::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 114 of file estimator.h.

Referenced by mappel::Estimator< Model >::clear_stats(), mappel::Estimator< Model >::get_stats(), and mappel::ThreadedEstimator< Model >::record_exit_code().

8.20.5.2 `template<class Model > int mappel::ThreadedEstimator< Model >::max_threads` [protected], [inherited]

Definition at line 142 of file estimator.h.

8.20.5.3 `template<class Model > Model& mappel::Estimator< Model >::model` [protected], [inherited]

Definition at line 109 of file estimator.h.

Referenced by mappel::SimulatedAnnealingMaximizer< Model >::anneal(), mappel::IterativeMaximizer< Model >::backtrack(), mappel::Estimator< Model >::compute_estimate(), mappel::CGaussMLE< Model >::compute_estimate(), mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate(), mappel::IterativeMaximizer< Model >::compute_estimate(), mappel::Estimator< Model >::compute_estimate_debug(), mappel::CGaussMLE< Model >::compute_estimate_debug(), mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug(), mappel::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::Estimator< Model >::compute_profile_estimate(), mappel::IterativeMaximizer< Model >::compute_profile_estimate(), mappel::Estimator< Model >::estimate_max(), mappel::Estimator< Model >::estimate_max_debug(), mappel::Estimator< Model >::estimate_max_stack(), mappel::ThreadedEstimator< Model >::estimate_max_stack(), mappel::ThreadedEstimator< Model >::estimate_profile_stack(), mappel::HeuristicEstimator< Model >::get_debug_stats(), mappel::Estimator< Model >::get_model(), mappel::IterativeMaximizer< Model >::local_maximize(), mappel::NewtonDiagonalMaximizer< Model >::maximize(), mappel::NewtonMaximizer< Model >::maximize(), mappel::QuasiNewtonMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >::maximize(), mappel::ThreadedEstimator< Model >::record_exit_code(), and mappel::Estimator< Model >::set_model().

8.20.5.4 `template<class Model > std::mutex mappel::ThreadedEstimator< Model >::mtx` [protected], [inherited]

Definition at line 144 of file estimator.h.

Referenced by mappel::ThreadedEstimator< Model >::clear_stats(), mappel::IterativeMaximizer< Model >::clear_stats(), mappel::ThreadedEstimator< Model >::get_stats(), mappel::IterativeMaximizer< Model >::get_stats(), mappel::ThreadedEstimator< Model >::record_exit_code(), and mappel::IterativeMaximizer< Model >::record_run_statistics().

8.20.5.5 `template<class Model> int mappel::Estimator< Model >::num_estimations = 0` [protected],
[inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

8.20.5.6 `template<class Model> int mappel::ThreadedEstimator< Model >::num_threads` [protected],
[inherited]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.20.5.7 `template<class Model> constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static],
[inherited]

Definition at line 39 of file estimator.h.

8.20.5.8 `template<class Model> double mappel::Estimator< Model >::total_walltime = 0.` [protected],
[inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

The documentation for this class was generated from the following files:

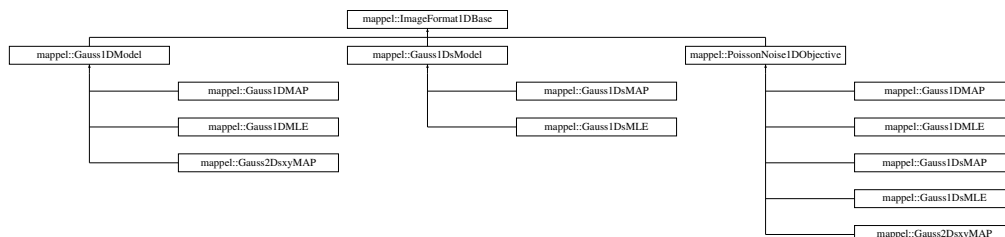
- [estimator.h](#)
- [estimator_impl.h](#)

8.21 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

8.21.1 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.

8.21.2 Member Typedef Documentation

8.21.2.1 using mappel::ImageFormat1DBase::ImageCoordT = uint32_t

Image size coordinate storage type

Definition at line 25 of file ImageFormat1DBase.h.

8.21.2.2 using mappel::ImageFormat1DBase::ImagePixelT = double

Image pixel storage type

Definition at line 26 of file ImageFormat1DBase.h.

8.21.2.3 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.

8.21.2.4 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.

8.21.2.5 using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.21.2.6 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.

8.21.2.7 using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.21.2.8 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.

8.21.2.9 using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.21.2.10 using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.21.3 Constructor & Destructor Documentation**8.21.3.1 mappel::ImageFormat1DBase::ImageFormat1DBase () [protected],[default]****8.21.3.2 ImageFormat1DBase::ImageFormat1DBase (ImageSizeT size_) [explicit],[protected]**

Definition at line 13 of file ImageFormat1DBase.cpp.

References check_size().

8.21.4 Member Function Documentation

8.21.4.1 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.

8.21.4.2 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.

8.21.4.3 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().

8.21.4.4 ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & *stack*, ImageCoordT *n*) const [inline]

Definition at line 108 of file ImageFormat1DBase.h.

8.21.4.5 ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_num_pixels () const [inline]

Definition at line 82 of file ImageFormat1DBase.h.

References size.

Referenced by get_stats().

8.21.4.6 ImageFormat1DBase::ImageSizeT ImageFormat1DBase::get_size () const [inline]

Definition at line 71 of file ImageFormat1DBase.h.

References size.

Referenced by get_stats().

8.21.4.7 **ImageFormat1DBase::ImageCoordT** ImageFormat1DBase::get_size (*IdxT idx*) const

Definition at line 20 of file ImageFormat1DBase.cpp.

References size.

8.21.4.8 **ImageFormat1DBase::ImageCoordT** ImageFormat1DBase::get_size_image_stack (const ImageStackT & *stack*) const [inline]

Definition at line 101 of file ImageFormat1DBase.h.

8.21.4.9 **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().

8.21.4.10 **ImageFormat1DBase::ImageT** ImageFormat1DBase::make_image () const [inline]

Definition at line 87 of file ImageFormat1DBase.h.

References size.

8.21.4.11 **ImageFormat1DBase::ImageStackT** ImageFormat1DBase::make_image_stack (ImageCoordT *n*) const [inline]

Definition at line 94 of file ImageFormat1DBase.h.

References size.

8.21.4.12 **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.

8.21.4.13 **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().

8.21.4.14 **void** ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & *sz*) [inline]

Definition at line 75 of file ImageFormat1DBase.h.

References set_size().

8.21.5 Member Data Documentation

8.21.5.1 `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()`.

8.21.5.2 `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()`.

8.21.5.3 `const ImageFormat1DBase::ImageCoordT ImageFormat1DBase::num_dim = 1` `[static]`

Number of image dimensions.

Definition at line 38 of file ImageFormat1DBase.h.

Referenced by `get_stats()`.

8.21.5.4 `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:

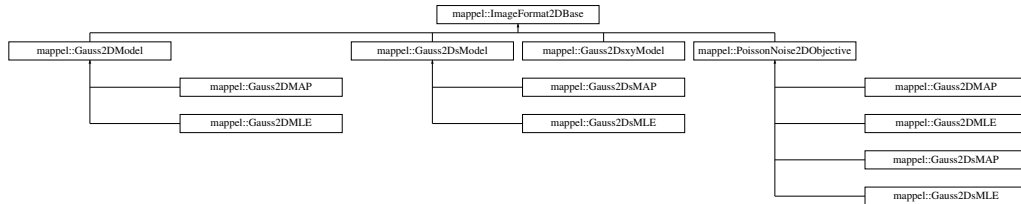
- [ImageFormat1DBase.h](#)
- [ImageFormat1DBase.cpp](#)

8.22 mappel::ImageFormat2DBase Class Reference

A virtual base class for 2D image localization objectives.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/ImageFormat2D←
Base.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

8.22.1 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](#).

8.22.2 Member Typedef Documentation

8.22.2.1 using mappel::ImageFormat2DBase::ImageCoordT = uint32_t

Image size coordinate storage type

Definition at line 24 of file [ImageFormat2DBase.h](#).

8.22.2.2 using mappel::ImageFormat2DBase::ImagePixelT = double

Image pixel storage type

Definition at line 25 of file [ImageFormat2DBase.h](#).

8.22.2.3 `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.

8.22.2.4 `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.

8.22.2.5 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

8.22.2.6 `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.

8.22.2.7 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.22.2.8 `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.

8.22.2.9 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.22.2.10 `using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT>`

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.22.3 Constructor & Destructor Documentation

8.22.3.1 `mappel::ImageFormat2DBase::ImageFormat2DBase ()` `[protected]`, `[default]`

8.22.3.2 `mappel::ImageFormat2DBase::ImageFormat2DBase (const ImageSizeT & size)` `[explicit]`, `[protected]`

Definition at line 13 of file ImageFormat2DBase.cpp.

References `check_size()`.

8.22.3.3 `mappel::ImageFormat2DBase::ImageFormat2DBase (const ImageFormat2DBase & o)` `[protected]`

Definition at line 19 of file ImageFormat2DBase.cpp.

8.22.3.4 `mappel::ImageFormat2DBase::ImageFormat2DBase (ImageFormat2DBase && o)` `[protected]`

Definition at line 23 of file ImageFormat2DBase.cpp.

8.22.4 Member Function Documentation

8.22.4.1 `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`.

8.22.4.2 `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`.

8.22.4.3 `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()`.

8.22.4.4 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const [inline]

Definition at line 106 of file ImageFormat2DBase.h.

8.22.4.5 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_num_pixels () const [inline]

Definition at line 79 of file ImageFormat2DBase.h.

References size.

Referenced by get_stats().

8.22.4.6 const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const [inline]

Definition at line 74 of file ImageFormat2DBase.h.

References size.

8.22.4.7 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT idx) const

Definition at line 41 of file ImageFormat2DBase.cpp.

References size.

8.22.4.8 ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & stack) const [inline]

Definition at line 99 of file ImageFormat2DBase.h.

8.22.4.9 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().

8.22.4.10 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const [inline]

Definition at line 85 of file ImageFormat2DBase.h.

References size.

8.22.4.11 ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT n) const [inline]

Definition at line 92 of file ImageFormat2DBase.h.

References size.

8.22.4.12 `ImageFormat2DBase & mappel::ImageFormat2DBase::operator= (const ImageFormat2DBase & o)`
`[protected]`

Definition at line 27 of file ImageFormat2DBase.cpp.

References size.

Referenced by `mappel::Gauss2DsMAP::operator=()`, `mappel::Gauss2DsMLE::operator=()`, `mappel::Gauss2DMAP::operator=()`, and `mappel::Gauss2DMLE::operator=()`.

8.22.4.13 `ImageFormat2DBase & mappel::ImageFormat2DBase::operator= (ImageFormat2DBase && o)`
`[protected]`

Definition at line 33 of file ImageFormat2DBase.cpp.

References size.

8.22.4.14 `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.

8.22.4.15 `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()`.

8.22.5 Member Data Documentation

8.22.5.1 `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()`.

8.22.5.2 `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()`.

8.22.5.3 `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()`.

8.22.5.4 `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()`.

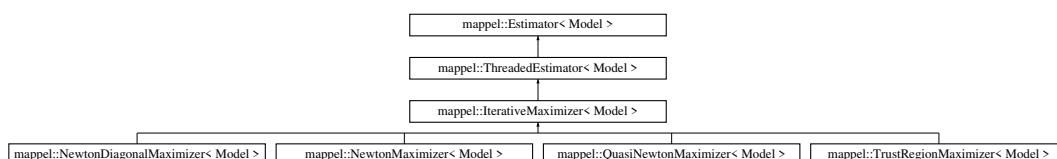
The documentation for this class was generated from the following files:

- [ImageFormat2DBase.h](#)
- [ImageFormat2DBase.cpp](#)

8.23 `mappel::IterativeMaximizer< Model >` Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::IterativeMaximizer< Model >`:



Classes

- class [MaximizerData](#)

Public Types

- enum [ExitCode](#) : IdxT {
[ExitCode::Unassigned](#) = 8, [ExitCode::MaxIter](#) = 7, [ExitCode::MaxBacktracks](#) = 6, [ExitCode::TrustRegionRadius](#) = 5,
[ExitCode::GradRatio](#) = 4, [ExitCode::FunctionChange](#) = 3, [ExitCode::StepSize](#) = 2, [ExitCode::Success](#) = 1,
[ExitCode::Error](#) = 0 }

Public Member Functions

- [IterativeMaximizer](#) (Model &model, int max_iterations=DEFAULT_ITERATIONS)
- double [mean_iterations](#) ()
- double [mean_backtracks](#) ()
- double [mean_fun_evals](#) ()
- double [mean_der_evals](#) ()
- [StatsT](#) [get_stats](#) ()
- [StatsT](#) [get_debug_stats](#) ()
- void [clear_stats](#) ()
- void [local_maximize](#) (const [ModelDataT](#)< Model > &im, const [StencilT](#)< Model > &theta_init, [StencilT](#)< Model > &stencil, double &rllh)
Perform a local maximization to finish off a simulated annealing run.
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im, const [ParamVecT](#)< Model > &theta_init, [ParamVecT](#)< Model > &theta, [VecT](#) &rllh, [CubeT](#) &obsI)
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im_stack, [ParamVecT](#)< Model > &theta_est, [VecT](#) &rllh_stack, [CubeT](#) &obsI_stack)
- void [estimate_profile_stack](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed_parameters, const [MatT](#) &values, const [ParamVecT](#)< Model > &theta_init, [VecT](#) &profile_likelihood, [ParamVecT](#)< Model > &profile_parameters)
- virtual std::string [name](#) () const =0
- Model & [get_model](#) ()
- void [set_model](#) (Model &new_model)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, double &rllh)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, double &rllh)
- void [estimate_max](#) (const [ModelDataT](#)< Model > &im, [ParamT](#)< Model > &theta, double &rllh, [MatT](#) &obsI)
- void [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta, double &rllh, [MatT](#) &obsI)
- void [estimate_max_debug](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsI, [MatT](#) &sequence, [VecT](#) &sequence_rllh)

Static Public Attributes

- static constexpr int [NumExitCodes](#) = 9

Protected Member Functions

- void [record_run_statistics](#) (const [MaximizerData](#) &data)
 - [StencilT](#)< [Model](#) > [compute_estimate](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, double &rllh)
 - [StencilT](#)< [Model](#) > [compute_estimate_debug](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, [ParamVecT](#)< [Model](#) > &sequence, [VecT](#) &sequence_rllh)
 - void [compute_profile_estimate](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, const [IdxVecT](#) &fixed_parameters, [ParamT](#)< [Model](#) > &theta_est, double &rllh)
 - virtual void [maximize](#) ([MaximizerData](#) &data)=0
 - bool [backtrack](#) ([MaximizerData](#) &data)
 - bool [convergence_test](#) ([MaximizerData](#) &data)
 - void [record_exit_code](#) ([ExitCode](#) code)
 - virtual void [compute_estimate](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, [ParamT](#)< [Model](#) > &theta_est, double &rllh, [MatT](#) &obsI)
- Default base class implementation computes rllh and obsI separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.*
- void [record_walltime](#) ([ClockT](#)::time_point start_walltime, int nimages)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = 1e-6
- double [delta](#) = 1e-6
- double [lambda_min](#) = 0.05
- double [alpha](#) = 1e-4
- int [max_backtracks](#) = 8
- int [total_iterations](#) = 0
- int [total_backtracks](#) = 0
- int [total_fun_evals](#) = 0
- int [total_der_evals](#) = 0
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- [Model](#) & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.
- [IdxVecT](#) [exit_counts](#)

8.23.1 Detailed Description

```
template<class Model>
class mappel::IterativeMaximizer< Model >
```

Definition at line 219 of file estimator.h.

8.23.2 Member Enumeration Documentation

8.23.2.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT` `[strong], [inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.23.3 Constructor & Destructor Documentation

8.23.3.1 `template<class Model > mappel::IterativeMaximizer< Model >::IterativeMaximizer (Model & model, int max_iterations = DEFAULT_ITERATIONS)`

Definition at line 525 of file estimator_impl.h.

8.23.4 Member Function Documentation

8.23.4.1 `template<class Model > bool mappel::IterativeMaximizer< Model >::backtrack (MaximizerData & data)` `[protected]`

Definition at line 647 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_`, `iteration()`, `mappel::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::step`.

Referenced by `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.23.4.2 `template<class Model> void mappel::IterativeMaximizer< Model>::clear_stats () [virtual]`

Reimplemented from [mappel::ThreadedEstimator< Model>](#).

Definition at line 625 of file `estimator_impl.h`.

References `mappel::ThreadedEstimator< Model>::clear_stats()`, `mappel::ThreadedEstimator< Model>::mtx`, `mappel::IterativeMaximizer< Model>::total_backtracks`, `mappel::IterativeMaximizer< Model>::total_der_evals`, `mappel::IterativeMaximizer< Model>::total_fun_evals`, and `mappel::IterativeMaximizer< Model>::total_iterations`.

8.23.4.3 `template<class Model> void mappel::Estimator< Model>::compute_estimate (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta_est, double & rllh, MatT & obsl) [protected],[virtual],[inherited]`

Default base class implementation computes `rllh` and `obsl` separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::model`, and `mappel::methods::observed_information()`.

8.23.4.4 `template<class Model> StencilT< Model> mappel::IterativeMaximizer< Model>::compute_estimate (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, double & rllh) [protected],[virtual]`

Implements [mappel::Estimator< Model>](#).

Definition at line 753 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model>::maximize()`, `mappel::Estimator< Model>::model`, `mappel::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::IterativeMaximizer< Model>::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model>::MaximizerData::stencil()`.

8.23.4.5 `template<class Model> StencilT< Model> mappel::IterativeMaximizer< Model>::compute_estimate_debug (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamVecT< Model> & sequence, VecT & sequence_rllh) [protected],[virtual]`

Estimators that produce a sequence of results (e.g. `IterativeEstimators`) can override this dummy debug implementation.

Reimplemented from [mappel::Estimator< Model>](#).

Definition at line 768 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh()`, `mappel::IterativeMaximizer< Model>::max_backtracks`, `mappel::IterativeMaximizer< Model>::max_iterations`, `mappel::IterativeMaximizer< Model>::maximize()`, `mappel::Estimator< Model>::model`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, `mappel::IterativeMaximizer< Model>::record_run_statistics()`, and `mappel::IterativeMaximizer< Model>::MaximizerData::stencil()`.

8.23.4.6 `template<class Model> void mappel::IterativeMaximizer< Model >::compute_profile_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT< Model > & theta_est, double & rllh) [protected], [virtual]`

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 797 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

8.23.4.7 `template<class Model> bool mappel::IterativeMaximizer< Model >::convergence_test (MaximizerData & data) [protected]`

Definition at line 731 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters](#), [mappel::IterativeMaximizer< Model >::MaximizerData::grad](#), [mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#).

8.23.4.8 `template<class Model> StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im) [inherited]`

Definition at line 64 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.23.4.9 `template<class Model> StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init) [inherited]`

Definition at line 82 of file estimator_impl.h.

References [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

8.23.4.10 `template<class Model> StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh) [inherited]`

Definition at line 73 of file estimator_impl.h.

References [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::model](#).

8.23.4.11 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [inherited]`

Definition at line 90 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.23.4.12 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 119 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.23.4.13 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

8.23.4.14 `template<class Model > void mappel::Estimator< Model >::estimate_max_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

Referenced by `mappel::Estimator< Model >::~Estimator()`.

8.23.4.15 `template<class Model > void mappel::Estimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im_stack, ParamVecT< Model > & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack) [inherited]`

Definition at line 170 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max_stack()`, and `mappel::Estimator< Model >::model`.

8.23.4.16 `template<class Model > void mappel::ThreadedEstimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im, const ParamVecT< Model > & theta_init, ParamVecT< Model > & theta, VecT & rllh, CubeT & obsl) [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 285 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::Error](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::num_threads](#), [mappel::print_image\(\)](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::Estimator< Model >::record_walltime\(\)](#), and [mappel::methods::objective::openmp::rllh_stack\(\)](#).

8.23.4.17 `template<class Model > void mappel::ThreadedEstimator< Model >::estimate_profile_stack (const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model > & theta_init, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters) [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 323 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_profile_estimate\(\)](#), [mappel::Estimator< Model >::Error](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::num_threads](#), [mappel::print_image\(\)](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), and [mappel::Estimator< Model >::record_walltime\(\)](#).

8.23.4.18 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_debug_stats () [virtual]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 609 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::backtrack_idx](#)s, [mappel::IterativeMaximizer< Model >::get_stats\(\)](#), and [mappel::IterativeMaximizer< Model >::last_backtrack_idx](#)s.

8.23.4.19 `template<class Model > Model & mappel::Estimator< Model >::get_model () [inherited]`

Definition at line 50 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.23.4.20 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_stats () [virtual]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 588 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::ThreadedEstimator< Model >::get_stats\(\)](#), [mappel::IterativeMaximizer< Model >::max_backtracks](#), [mappel::IterativeMaximizer< Model >::max_iterations](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::Estimator< Model >::num_estimations](#), [mappel::IterativeMaximizer< Model >::total_backtracks](#), [mappel::IterativeMaximizer< Model >::total_der_evals](#), [mappel::IterativeMaximizer< Model >::total_fun_evals](#), and [mappel::IterativeMaximizer< Model >::total_iterations](#).

Referenced by [mappel::SimulatedAnnealingMaximizer< Model >::get_debug_stats\(\)](#), and [mappel::IterativeMaximizer< Model >::get_debug_stats\(\)](#).

8.23.4.21 `template<class Model > void mappel::IterativeMaximizer< Model >::local_maximize (const ModelDataT< Model > & im, const StencilT< Model > & theta_init, StencilT< Model > & stencil, double & rllh)`

Perform a local maximization to finish off a simulated annealing run.

Definition at line 811 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`.

8.23.4.22 `template<class Model > virtual void mappel::IterativeMaximizer< Model >::maximize (MaximizerData & data) [protected], [pure virtual]`

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, and `mappel::IterativeMaximizer< Model >::local_maximize()`.

8.23.4.23 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_backtracks ()`

8.23.4.24 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_der_evals ()`

8.23.4.25 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_fun_evals ()`

8.23.4.26 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_iterations ()`

8.23.4.27 `template<class Model > virtual std::string mappel::Estimator< Model >::name () const [pure virtual], [inherited]`

Implemented in `mappel::TrustRegionMaximizer< Model >`, `mappel::QuasiNewtonMaximizer< Model >`, `mappel::NewtonMaximizer< Model >`, `mappel::NewtonDiagonalMaximizer< Model >`, `mappel::SimulatedAnnealingMaximizer< Model >`, `mappel::CGaussMLE< Model >`, `mappel::CGaussHeuristicEstimator< Model >`, and `mappel::HeuristicEstimator< Model >`.

Referenced by `mappel::Estimator< Model >::~Estimator()`.

8.23.4.28 `template<class Model > void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code) [protected], [virtual], [inherited]`

Implements `mappel::Estimator< Model >`.

Definition at line 386 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::exit_counts`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::mtx`, and `mappel::methods::objective::rllh()`.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::maximize()`.

8.23.4.29 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const MaximizerData & data)` [protected]

Definition at line 636 of file estimator_impl.h.

References mappel::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idx(), mappel::IterativeMaximizer< Model >::last_backtrack_idx(), mappel::ThreadedEstimator< Model >::mtx, mappel::IterativeMaximizer< Model >::MaximizerData::nBacktracks, mappel::IterativeMaximizer< Model >::MaximizerData::nIterations, mappel::IterativeMaximizer< Model >::MaximizerData::save_seq, mappel::IterativeMaximizer< Model >::total_backtracks, mappel::IterativeMaximizer< Model >::total_der_evals, mappel::IterativeMaximizer< Model >::total_fun_evals, and mappel::IterativeMaximizer< Model >::total_iterations.

Referenced by mappel::IterativeMaximizer< Model >::compute_estimate(), mappel::IterativeMaximizer< Model >::compute_estimate_debug(), and mappel::IterativeMaximizer< Model >::compute_profile_estimate().

8.23.4.30 `template<class Model > void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages)` [protected],[inherited]

Definition at line 266 of file estimator_impl.h.

References mappel::Estimator< Model >::num_estimations, and mappel::Estimator< Model >::total_walltime.

Referenced by mappel::Estimator< Model >::estimate_max(), mappel::Estimator< Model >::estimate_max_debug(), mappel::ThreadedEstimator< Model >::estimate_max_stack(), and mappel::ThreadedEstimator< Model >::estimate_profile_stack().

8.23.4.31 `template<class Model > void mappel::Estimator< Model >::set_model (Model & new_model)` [inherited]

Definition at line 54 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::~Estimator().

8.23.5 Member Data Documentation

8.23.5.1 `template<class Model > double mappel::IterativeMaximizer< Model >::alpha = 1e-4` [protected]

Definition at line 245 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::backtrack(), and mappel::TrustRegionMaximizer< Model >::bound_step().

8.23.5.2 `template<class Model > double mappel::IterativeMaximizer< Model >::delta = 1e-6` [protected]

Definition at line 242 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::convergence_test(), mappel::IterativeMaximizer< Model >::get_stats(), mappel::NewtonMaximizer< Model >::maximize(), mappel::QuasiNewtonMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton(), and mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem().

8.23.5.3 `template<class Model> double mappel::IterativeMaximizer< Model >::epsilon = 1e-6` [protected]

Definition at line 241 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`.

8.23.5.4 `template<class Model> IdxVecT mappel::Estimator< Model >::exit_counts` [protected],
[inherited]

Definition at line 114 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, and `mappel::ThreadedEstimator< Model >::record_exit_code()`.

8.23.5.5 `template<class Model> double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` [protected]

Definition at line 244 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`, and `mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem()`.

8.23.5.6 `template<class Model> IdxVecT mappel::IterativeMaximizer< Model >::last_backtrack_idx`s [protected]

Definition at line 255 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.23.5.7 `template<class Model> int mappel::IterativeMaximizer< Model >::max_backtracks = 8` [protected]

Definition at line 246 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, and `mappel::IterativeMaximizer< Model >::get_stats()`.

8.23.5.8 `template<class Model> int mappel::IterativeMaximizer< Model >::max_iterations` [protected]

Definition at line 238 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.23.5.9 `template<class Model> int mappel::ThreadedEstimator< Model >::max_threads` [protected],
[inherited]

Definition at line 142 of file estimator.h.

8.23.5.10 `template<class Model > Model& mappel::Estimator< Model >::model` [protected], [inherited]

Definition at line 109 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::Estimator< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::estimate_max()`, `mappel::Estimator< Model >::estimate_max_debug()`, `mappel::Estimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::Estimator< Model >::get_model()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::set_model()`.

8.23.5.11 `template<class Model > std::mutex mappel::ThreadedEstimator< Model >::mtx` [protected], [inherited]

Definition at line 144 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.23.5.12 `template<class Model > int mappel::Estimator< Model >::num_estimations = 0` [protected], [inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

8.23.5.13 `template<class Model > int mappel::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.23.5.14 `template<class Model > constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static], [inherited]

Definition at line 39 of file estimator.h.

8.23.5.15 `template<class Model> int mappel::IterativeMaximizer< Model>::total_backtracks = 0` [protected]

Definition at line 250 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::clear_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, and `mappel::IterativeMaximizer< Model>::record_run_statistics()`.

8.23.5.16 `template<class Model> int mappel::IterativeMaximizer< Model>::total_der_evals = 0` [protected]

Definition at line 252 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::clear_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, and `mappel::IterativeMaximizer< Model>::record_run_statistics()`.

8.23.5.17 `template<class Model> int mappel::IterativeMaximizer< Model>::total_fun_evals = 0` [protected]

Definition at line 251 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::clear_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, and `mappel::IterativeMaximizer< Model>::record_run_statistics()`.

8.23.5.18 `template<class Model> int mappel::IterativeMaximizer< Model>::total_iterations = 0` [protected]

Definition at line 249 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::clear_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, and `mappel::IterativeMaximizer< Model>::record_run_statistics()`.

8.23.5.19 `template<class Model> double mappel::Estimator< Model>::total_walltime = 0.` [protected], [inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model>::clear_stats()`, `mappel::Estimator< Model>::get_stats()`, `mappel::ThreadedEstimator< Model>::get_stats()`, and `mappel::Estimator< Model>::record_walltime()`.

The documentation for this class was generated from the following files:

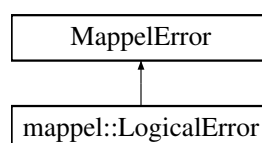
- [estimator.h](#)
- [estimator_impl.h](#)

8.24 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)

8.24.1 Detailed Description

Failure of code or algorithm logic.

Definition at line 100 of file util.h.

8.24.2 Constructor & Destructor Documentation

8.24.2.1 mappel::LogicalError::LogicalError (std::string *message*) [inline]

Definition at line 102 of file util.h.

The documentation for this struct was generated from the following file:

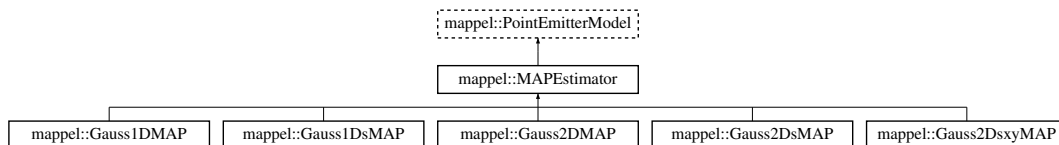
- [util.h](#)

8.25 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)
- [ParamT sample_prior](#) ()
- void [set_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set_lbound](#) (const [ParamT](#) &lbound)
- void [set_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get_lbound](#) () const
- const [ParamT](#) & [get_ubound](#) () const
- bool [theta_in_bounds](#) (const [ParamT](#) &theta) const
- void [bound_theta](#) ([ParamT](#) &theta, double epsilon=[bounds_epsilon](#)) const
- [ParamT bounded_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds_epsilon](#)) const
- [ParamT reflected_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT theta_stack_in_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT bounded_theta_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds_epsilon](#)) const
- [ParamVecT reflected_theta_stack](#) (const [ParamVecT](#) &theta) const

Static Public Member Functions

- static prior_hessian::TruncatedNormalDist [make_prior_component_position_normal](#) (IdxT size, double pos_↵, double sigma=[default_sigma_pos](#))
- static prior_hessian::ScaledSymmetricBetaDist [make_prior_component_position_beta](#) (IdxT size, double pos_↵, double beta=[default_beta_pos](#))
- static prior_hessian::TruncatedGammaDist [make_prior_component_intensity](#) (double mean=[default_mean_l](#), double kappa=[default_intensity_kappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double min_sigma, double max_↵, double sigma, double alpha=[default_alpha_sigma](#))
- static void [set_rng_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()

Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2

Protected Member Functions

- [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](#)

8.25.1 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.

8.25.2 Member Typedef Documentation

8.25.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.25.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.25.3 Constructor & Destructor Documentation

8.25.3.1 `mappel::MAPEstimator::MAPEstimator ()` [inline],[protected]

Definition at line 24 of file MAPEstimator.h.

8.25.3.2 `mappel::MAPEstimator::MAPEstimator (const MAPEstimator & o)` [inline],[protected]

Definition at line 25 of file MAPEstimator.h.

8.25.3.3 `mappel::MAPEstimator::MAPEstimator (MAPEstimator && o)` [inline],[protected]

Definition at line 26 of file MAPEstimator.h.

8.25.4 Member Function Documentation

8.25.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const`
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.25.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded_theta_stack().

8.25.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::make_param_stack().

8.25.4.4 void mappel::PointEmitterModel::check_param_shape (const ParamT & *theta*) const [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.25.4.5 void mappel::PointEmitterModel::check_param_shape (const ParamVecT & *theta*) const [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.25.4.6 void mappel::PointEmitterModel::check_psf_sigma (double *psf_sigma*) const [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.25.4.7 void mappel::PointEmitterModel::check_psf_sigma (const VecT & *psf_sigma*) const [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.25.4.8 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.25.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`,`[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.25.4.10 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.25.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.25.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.25.4.13 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`,`[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.25.4.14 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`,`[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.25.4.15 StringVecT mappel::PointEmitterModel::get_param_names () const [inline],[inherited]

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.25.4.16 CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

8.25.4.17 const CompositeDist & mappel::PointEmitterModel::get_prior () const [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.25.4.18 ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.25.4.19 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.25.4.20 StatsT mappel::PointEmitterModel::get_stats () const [inherited]

Definition at line 125 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_hyperparams, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, mappel::rng_manager, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DModel::get_stats(), mappel::Gauss1DsModel::get_stats(), mappel::Gauss2DModel::get_stats(), and mappel::Gauss2DsModel::get_stats().

8.25.4.21 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.25.4.22 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const` `[inline],[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.25.4.23 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const` `[inline],[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.25.4.24 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const` `[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.25.4.25 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.25.4.26 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.25.4.27 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.25.4.28 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.25.4.29 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline]`,
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

8.25.4.30 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.25.4.31 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_I, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::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()`.

8.25.4.32 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` `[static]`, `[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

8.25.4.33 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal (IdxT size, double pos_sigma = default_sigma_pos)` `[static]`, `[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.25.4.34 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma (double min_sigma, double max_sigma, double alpha = default_alpha_sigma)` `[static],[inherited]`

Definition at line 104 of file `PointEmitterModel.cpp`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.25.4.35 `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=()`.

8.25.4.36 `MAPEstimator& mappel::MAPEstimator::operator= (MAPEstimator && o)` `[inline],[protected]`

Definition at line 28 of file `MAPEstimator.h`.

8.25.4.37 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const` `[inherited]`

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

8.25.4.38 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const` `[inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.25.4.39 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)` `[inline],[inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.25.4.40 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.25.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ()` `[inline]`, `[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.25.4.42 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.25.4.43 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc)` `[inline]`,
`[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.25.4.44 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` `[inline]`,
`[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.25.4.45 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` `[inline]`, `[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

8.25.4.46 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.25.4.47 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc)` `[inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.25.4.48 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

8.25.4.49 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.25.4.50 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` `[static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.25.4.51 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

8.25.4.52 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const` `[inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.25.4.53 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const` `[inherited]`

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.25.5 Member Data Documentation

8.25.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.25.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.25.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.25.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.25.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

8.25.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.25.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.25.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.25.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

8.25.5.10 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.25.5.11 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.25.5.12 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.25.5.13 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

8.25.5.14 `IdxT mappel::PointEmitterModel::num_params` `[protected], [inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.25.5.15 `CompositeDist mappel::PointEmitterModel::prior` `[protected], [inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.25.5.16 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

The documentation for this class was generated from the following file:

- [MAPEstimator.h](#)

8.26 mappel::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, int max_seq_len=0)
- 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)
- [ParamVecT](#)< Model > [get_theta_sequence](#) () const
Return the saved theta sequence.
- [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.
- [StencilT](#)< Model > & [saved_stencil](#) ()
Get the saved stencil.
- [ParamT](#)< Model > & [theta](#) ()
Get the current stencil's theta.
- [ParamT](#)< Model > & [saved_theta](#) ()
Get the saved stencil's theta.
- int [getIteration](#) () const
- void [set_fixed_parameters](#) (const [IdxVecT](#) &fixed_parameters)

Public Attributes

- const [ModelDataT](#)< Model > & [im](#)
- [ParamT](#)< Model > [grad](#)
- [ParamT](#)< Model > [step](#)
- [VecT](#) [lbound](#)
- [VecT](#) [ubound](#)
- double [rllh](#)
- int [nBacktracks](#) =0
- int [nIterations](#) =0
- bool [save_seq](#)
- [VecT](#) [free_parameters](#)
- [IdxT](#) [num_fixed_parameters](#) =0
- [IdxVecT](#) [fixed_idx](#)s

Protected Attributes

- [StencilT](#)< Model > [s0](#)
- [StencilT](#)< Model > [s1](#)
- bool [current_stencil](#)
- [ParamVecT](#)< Model > [theta_seq](#)
- [VecT](#) [seq_rllh](#)
- [IdxVecT](#) [backtrack_idx](#)s
- int [seq_len](#) =0
- const int [max_seq_len](#)

8.26.1 Detailed Description

```
template<class Model>
class mappel::IterativeMaximizer< Model >::MaximizerData
```

Definition at line 257 of file estimator.h.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 `template<class Model> mappel::IterativeMaximizer< Model >::MaximizerData (const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & s, bool save_seq = false, int max_seq_len = 0)`

Definition at line 531 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::backtrack_idx](#)s, [mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::seq_rllh](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta_seq](#).

8.26.3 Member Function Documentation

8.26.3.1 `template<class Model> IdxVecT mappel::IterativeMaximizer< Model>::MaximizerData::get_backtrack_idx() const [inline]`

Definition at line 280 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::record_run_statistics()`.

8.26.3.2 `template<class Model> ParamVecT<Model> mappel::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence() const [inline]`

Return the saved theta sequence.

Definition at line 279 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`.

8.26.3.3 `template<class Model> VecT mappel::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence_rllh() const [inline]`

Definition at line 281 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`.

8.26.3.4 `template<class Model> int mappel::IterativeMaximizer< Model>::MaximizerData::getIteration() const [inline]`

Definition at line 301 of file estimator.h.

8.26.3.5 `template<class Model> void mappel::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 274 of file estimator.h.

References `mappel::IterativeMaximizer< Model>::MaximizerData::record_backtrack()`.

Referenced by `mappel::IterativeMaximizer< Model>::backtrack()`, and `mappel::IterativeMaximizer< Model>::MaximizerData::record_backtrack()`.

8.26.3.6 `template<class Model> void mappel::IterativeMaximizer< Model>::MaximizerData::record_backtrack(const ParamT< Model> & rejected_theta, double rejected_rllh)`

Record a backtracked point (no derivative computations performed)

Definition at line 566 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model>::MaximizerData::backtrack_idx`, `mappel::IterativeMaximizer< Model>::MaximizerData::max_seq_len`, `mappel::IterativeMaximizer< Model>::MaximizerData::nBacktracks`, `mappel::IterativeMaximizer< Model>::MaximizerData::save_seq`, `mappel::IterativeMaximizer< Model>::MaximizerData::seq_len`, `mappel::IterativeMaximizer< Model>::MaximizerData::seq_rllh`, and `mappel::IterativeMaximizer< Model>::MaximizerData::theta_seq`.

8.26.3.7 `template<class Model > void mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration ()`
`[inline]`

Definition at line 270 of file estimator.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::MaximizerData::MaximizerData()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

8.26.3.8 `template<class Model > void mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration (const ParamT< Model > & accepted_theta)`

Record an iteration point (derivatives computed)

Definition at line 554 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::max_seq_len`, `mappel::IterativeMaximizer< Model >::MaximizerData::nIterations`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::save_seq`, `mappel::IterativeMaximizer< Model >::MaximizerData::seq_len`, `mappel::IterativeMaximizer< Model >::MaximizerData::seq_rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::theta_seq`.

8.26.3.9 `template<class Model > void mappel::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 294 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`.

8.26.3.10 `template<class Model > void mappel::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 289 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`.

8.26.3.11 `template<class Model > StencilT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::saved_stencil ()` `[inline]`

Get the saved stencil.

Definition at line 296 of file estimator.h.

8.26.3.12 `template<class Model > ParamT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta () [inline]`

Get the saved stencil's theta.

Definition at line 300 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, and `mappel::IterativeMaximizer< Model >::convergence_test()`.

8.26.3.13 `template<class Model > void mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters (const IdxVecT & fixed_parameters)`

Definition at line 579 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_idxes`, `mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters`, and `mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters`.

Referenced by `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`.

8.26.3.14 `template<class Model > void mappel::IterativeMaximizer< Model >::MaximizerData::set_stencil (const StencilT< Model > & s) [inline]`

Definition at line 284 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`.

8.26.3.15 `template<class Model > StencilT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::stencil () [inline]`

Get the current stencil.

Definition at line 283 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, and `mappel::IterativeMaximizer< Model >::local_maximize()`.

8.26.3.16 `template<class Model > ParamT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::theta () [inline]`

Get the current stencil's theta.

Definition at line 298 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::compute_bound_scaling_vec()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, and `mappel::IterativeMaximizer< Model >::convergence_test()`.

8.26.4 Member Data Documentation

8.26.4.1 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`
`[protected]`

Definition at line 313 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::get_debug_stats()`, `mappel::IterativeMaximizer< Model >::MaximizerData::MaximizerData()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`.

8.26.4.2 `template<class Model > bool mappel::IterativeMaximizer< Model >::MaximizerData::current_stencil`
`[protected]`

Definition at line 309 of file estimator.h.

8.26.4.3 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::MaximizerData::fixed_idx`

Definition at line 305 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`.

8.26.4.4 `template<class Model > VecT mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters`

Definition at line 303 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::convergence_test()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`.

8.26.4.5 `template<class Model > ParamT<Model> mappel::IterativeMaximizer< Model >::MaximizerData::grad`

Definition at line 260 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, and `mappel::IterativeMaximizer< Model >::convergence_test()`.

8.26.4.6 `template<class Model > const ModelDataT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::im`

Definition at line 259 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`.

8.26.4.7 `template<class Model > VecT mappel::IterativeMaximizer< Model >::MaximizerData::lbound`

Definition at line 262 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::compute_bound_scaling_vec()`.

8.26.4.8 `template<class Model > const int mappel::IterativeMaximizer< Model >::MaximizerData::max_seq_len`
`[protected]`

Definition at line 315 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

8.26.4.9 `template<class Model > int mappel::IterativeMaximizer< Model >::MaximizerData::nBacktracks =0`

Definition at line 264 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.26.4.10 `template<class Model > int mappel::IterativeMaximizer< Model >::MaximizerData::nIterations =0`

Definition at line 265 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.26.4.11 `template<class Model > IdxT mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters =0`

Definition at line 304 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::convergence_test()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters()`.

8.26.4.12 `template<class Model > double mappel::IterativeMaximizer< Model >::MaximizerData::rllh`

Definition at line 263 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

8.26.4.13 `template<class Model > StencilT<Model> mappel::IterativeMaximizer< Model >::MaximizerData::s0`
`[protected]`

Definition at line 308 of file estimator.h.

8.26.4.14 `template<class Model > StencilT<Model> mappel::IterativeMaximizer< Model >::MaximizerData::s1`
`[protected]`

Definition at line 308 of file estimator.h.

8.26.4.15 `template<class Model > bool mappel::IterativeMaximizer< Model >::MaximizerData::save_seq`

Definition at line 266 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.26.4.16 `template<class Model > int mappel::IterativeMaximizer< Model >::MaximizerData::seq_len=0` [protected]

Definition at line 314 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

8.26.4.17 `template<class Model > VecT mappel::IterativeMaximizer< Model >::MaximizerData::seq_rllh` [protected]

Definition at line 312 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::MaximizerData()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

8.26.4.18 `template<class Model > ParamT<Model> mappel::IterativeMaximizer< Model >::MaximizerData::step`

Definition at line 261 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, and `mappel::TrustRegionMaximizer< Model >::bound_step()`.

8.26.4.19 `template<class Model > ParamVecT<Model> mappel::IterativeMaximizer< Model >::MaximizerData::theta_seq` [protected]

Definition at line 311 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::MaximizerData()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

8.26.4.20 `template<class Model > VecT mappel::IterativeMaximizer< Model >::MaximizerData::ubound`

Definition at line 262 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::compute_bound_scaling_vec()`.

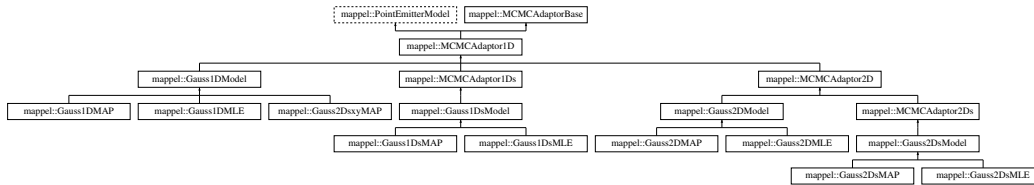
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

8.27 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)
- 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)
- [ParamT](#) [sample_prior](#) ()
- void [set_bounds](#) (const [ParamT](#) &lbound, const [ParamT](#) &ubound)
- void [set_lbound](#) (const [ParamT](#) &lbound)
- void [set_ubound](#) (const [ParamT](#) &ubound)
- const [ParamT](#) & [get_lbound](#) () const
- const [ParamT](#) & [get_ubound](#) () const
- bool [theta_in_bounds](#) (const [ParamT](#) &theta) const
- void [bound_theta](#) ([ParamT](#) &theta, double epsilon=[bounds_epsilon](#)) const
- [ParamT](#) [bounded_theta](#) (const [ParamT](#) &theta, double epsilon=[bounds_epsilon](#)) const
- [ParamT](#) [reflected_theta](#) (const [ParamT](#) &theta) const
- [BoolVecT](#) [theta_stack_in_bounds](#) (const [ParamVecT](#) &theta) const
- [ParamVecT](#) [bounded_theta_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds_epsilon](#)) const
- [ParamVecT](#) [reflected_theta_stack](#) (const [ParamVecT](#) &theta) const
- void [set_mcmc_sigma_scale](#) (double scale)
- double [get_mcmc_sigma_scale](#) () const
- [IdxT](#) [get_mcmc_num_phases](#) () const

Static Public Member Functions

- static [prior_hessian::TruncatedNormalDist](#) [make_prior_component_position_normal](#) ([IdxT](#) size, double pos_↵, double sigma=[default_sigma_pos](#))
- static [prior_hessian::ScaledSymmetricBetaDist](#) [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↵, double beta=[default_beta_pos](#))
- static [prior_hessian::TruncatedGammaDist](#) [make_prior_component_intensity](#) (double mean=[default_mean_↵](#), double kappa=[default_intensity_kappa](#))
- static [prior_hessian::TruncatedParetoDist](#) [make_prior_component_sigma](#) (double min_sigma, double max_↵, double sigma, double alpha=[default_alpha_sigma](#))
- static void [set_rng_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()

Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300

- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2
- 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`

8.27.1 Detailed Description

Definition at line 15 of file `MCMCAdaptor1D.h`.

8.27.2 Member Typedef Documentation

8.27.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file `PointEmitterModel.h`.

8.27.2.2 using mappel::PointEmitterModel::ParamVecT = arma::mat [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.27.3 Constructor & Destructor Documentation

8.27.3.1 mappel::MCMCAdaptor1D::MCMCAdaptor1D () [protected]

Definition at line 11 of file MCMCAdaptor1D.cpp.

8.27.3.2 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`.

8.27.3.3 mappel::MCMCAdaptor1D::MCMCAdaptor1D (const MCMCAdaptor1D & o) [protected]

Definition at line 24 of file MCMCAdaptor1D.cpp.

References `eta_bg`, `eta_l`, and `eta_x`.

8.27.3.4 mappel::MCMCAdaptor1D::MCMCAdaptor1D (MCMCAdaptor1D && o) [protected]

Definition at line 33 of file MCMCAdaptor1D.cpp.

References `eta_bg`, `eta_l`, and `eta_x`.

8.27.4 Member Function Documentation

8.27.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::Point↔EmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.27.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::Point↔EmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

8.27.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.27.4.4 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.27.4.5 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.27.4.6 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.27.4.7 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.27.4.8 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` [inline], [inherited]

Definition at line 236 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.27.4.9 StringVecT mappel::PointEmitterModel::get_hyperparam_names () const [inline],[inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.27.4.10 double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const [inline],[inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by set_background_mcmc_sampling(), and set_intensity_mcmc_sampling().

8.27.4.11 PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const [inline],[inherited]

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.27.4.12 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const [inline],[inherited]

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and set_background_mcmc_sampling().

8.27.4.13 IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const [inherited]

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.27.4.14 double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const [inherited]

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.27.4.15 IdxT mappel::PointEmitterModel::get_num_hyperparams () const [inline],[inherited]

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.27.4.16 `IdxT mappel::PointEmitterModel::get_num_params () const [inline],[inherited]`

Definition at line 160 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.27.4.17 `StringVecT mappel::PointEmitterModel::get_param_names () const [inline],[inherited]`

Definition at line 248 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.27.4.18 `CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]`

Definition at line 200 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

8.27.4.19 `const CompositeDist & mappel::PointEmitterModel::get_prior () const [inline],[inherited]`

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.27.4.20 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.27.4.21 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]`

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.27.4.22 `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()`.

8.27.4.23 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const` `[inline]`,
`[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.27.4.24 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const` `[inline]`,`[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.27.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const` `[inline]`,`[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.27.4.26 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const`
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.27.4.27 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline]`,`[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.27.4.28 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.27.4.29 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.27.4.30 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.27.4.31 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.27.4.32 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.27.4.33 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.27.4.34 **prior_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make_prior_component_position_beta (*IdxT size*, *double pos_beta = default_beta_pos*) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), and mappel::Gauss2DsModel::make_prior_beta_position().

8.27.4.35 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (*IdxT size*, *double pos_sigma = default_sigma_pos*) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss2DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.27.4.36 **prior_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make_prior_component_sigma (*double min_sigma*, *double max_sigma*, *double alpha = default_alpha_sigma*) [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.27.4.37 **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=().

8.27.4.38 **MCMCAdaptor1D & mappel::MCMCAdaptor1D::operator=** (*MCMCAdaptor1D && o*) [protected]

Definition at line 53 of file MCMCAdaptor1D.cpp.

References eta_bg, eta_l, and eta_x.

8.27.4.39 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const`
`[inherited]`

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

8.27.4.40 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const`
`[inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.27.4.41 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)`
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.27.4.42 `void mappel::MCMCAdaptor1D::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)`

Definition at line 108 of file `MCMCAdaptor1D.cpp`.

References `eta_bg`, `eta_l`, `eta_x`, `mappel::MCMCAdaptorBase::num_phases`, and `mappel::rng_manager`.

8.27.4.43 `template<class RngT> PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.27.4.44 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ()` `[inline], [inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.27.4.45 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `MCMCAdaptor1D()`.

8.27.4.46 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`
[*inherited*]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.27.4.47 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc)` [*inline*],
[*inherited*]

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.27.4.48 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` [*inline*],
[*inherited*]

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.27.4.49 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` [*inline*], [*inherited*]

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

8.27.4.50 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1)`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `MCMCAdaptor1D()`.

8.27.4.51 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.27.4.52 `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()`.

8.27.4.53 `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`.

8.27.4.54 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc)` `[inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.27.4.55 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

8.27.4.56 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.27.4.57 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.27.4.58 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set_max_sigma(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.27.4.59 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::make_stencil(), mappel::Gauss1DsModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DsxModel::make_stencil(), mappel::Gauss2DsModel::make_stencil(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.27.4.60 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const [inherited]`

Definition at line 297 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

8.27.5 Member Data Documentation

8.27.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static],[inherited]`

Distance from the boundary to constrain in bound_theta and bounded_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss2DsModel::set_max_sigma_ratio(), and mappel::PointEmitterModel::set_ubound().

8.27.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2 [static],[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.27.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.27.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.27.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

8.27.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `set_intensity_mcmc_sampling()`.

8.27.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `set_background_mcmc_sampling()`.

8.27.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.27.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static],`
`[inherited]`

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

8.27.5.10 `double mappel::MCMCAdaptor1D::eta_bg = 0 [protected]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.27.5.11 `double mappel::MCMCAdaptor1D::eta_l = 0 [protected]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.27.5.12 `double mappel::MCMCAdaptor1D::eta_x = 0 [protected]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.27.5.13 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05 [static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.27.5.14 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5 [static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.27.5.15 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static],[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.27.5.16 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static],[inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.27.5.17 `ParamT mappel::PointEmitterModel::lbound` `[protected],[inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.27.5.18 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected],[inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

8.27.5.19 `IdxT mappel::PointEmitterModel::num_params` `[protected],[inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.27.5.20 IdxT mappel::MCMCAdaptorBase::num_phases [protected],[inherited]

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_num_phases(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor2Ds::sample_mcmc_candidate(), mappel::MCMCAdaptor2D::sample_mcmc_candidate(), mappel::MCMCAdaptor1Ds::sample_mcmc_candidate(), sample_mcmc_candidate(), and mappel::MCMCAdaptorBase::set_mcmc_num_phases().

8.27.5.21 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get_hyperparam_index(), mappel::PointEmitterModel::get_hyperparam_names(), mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_hyperparams(), mappel::Gauss1DsModel::get_max_sigma(), mappel::Gauss1DsModel::get_min_sigma(), mappel::PointEmitterModel::get_param_names(), mappel::PointEmitterModel::get_prior(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::has_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename_hyperparam(), mappel::PointEmitterModel::sample_prior(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_hyperparam_names(), mappel::PointEmitterModel::set_hyperparam_value(), mappel::PointEmitterModel::set_hyperparams(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::PointEmitterModel::set_param_names(), mappel::PointEmitterModel::set_prior(), and mappel::PointEmitterModel::set_ubound().

8.27.5.22 double mappel::MCMCAdaptorBase::sigma_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_sigma_scale(), mappel::MCMCAdaptorBase::get_stats(), MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds(), set_background_mcmc_sampling(), set_intensity_mcmc_sampling(), and mappel::MCMCAdaptorBase::set_mcmc_sigma_scale().

8.27.5.23 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

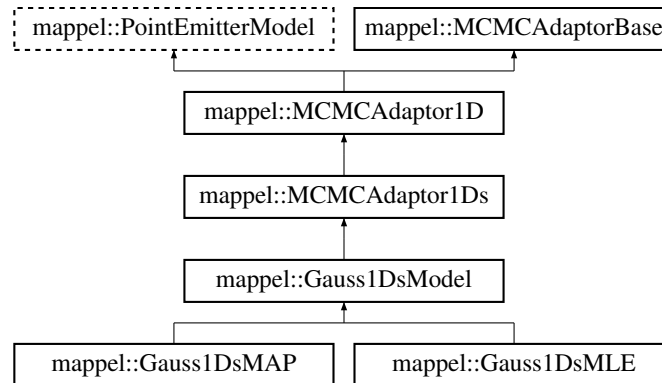
The documentation for this class was generated from the following files:

- [MCMCAdaptor1D.h](#)
- [MCMCAdaptor1D.cpp](#)

8.28 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)
- 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)
- `ParamT` `sample_prior` ()
- void `set_bounds` (const `ParamT` &lbound, const `ParamT` &ubound)
- void `set_lbound` (const `ParamT` &lbound)
- void `set_ubound` (const `ParamT` &ubound)
- const `ParamT` & `get_lbound` () const
- const `ParamT` & `get_ubound` () const
- bool `theta_in_bounds` (const `ParamT` &theta) const
- void `bound_theta` (`ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `bounded_theta` (const `ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `reflected_theta` (const `ParamT` &theta) const
- `BoolVecT` `theta_stack_in_bounds` (const `ParamVecT` &theta) const
- `ParamVecT` `bounded_theta_stack` (const `ParamVecT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamVecT` `reflected_theta_stack` (const `ParamVecT` &theta) const
- void `set_mcmc_sigma_scale` (double scale)
- double `get_mcmc_sigma_scale` () const
- `IdxT` `get_mcmc_num_phases` () const

Static Public Member Functions

- static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` size, double pos_↵
sigma=`default_sigma_pos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos_↵
beta=`default_beta_pos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=`default_mean_↵`
l, double kappa=`default_intensity_kappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min_sigma, double max_↵
sigma, double alpha=`default_alpha_sigma`)
- static void `set_rng_seed` (`RngSeedT` seed)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5

Protected Member Functions

- [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](#)

8.28.1 Detailed Description

Definition at line 14 of file [MCMCAdaptor1Ds.h](#).

8.28.2 Member Typedef Documentation

8.28.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.28.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.28.3 Constructor & Destructor Documentation

8.28.3.1 `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds ()` [protected]

Definition at line 11 of file MCMCAdaptor1Ds.cpp.

References `mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale`.

8.28.3.2 `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`.

8.28.3.3 `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds (const MCMCAdaptor1Ds & o)` [protected]

Definition at line 21 of file MCMCAdaptor1Ds.cpp.

References `eta_sigma`.

8.28.3.4 `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds (MCMCAdaptor1Ds && o)` [protected]

Definition at line 28 of file MCMCAdaptor1Ds.cpp.

References `eta_sigma`.

8.28.4 Member Function Documentation

8.28.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const`
[inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.28.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

8.28.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.28.4.4 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.28.4.5 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.28.4.6 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.28.4.7 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.28.4.8 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.28.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`,`[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.28.4.10 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.28.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.28.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.28.4.13 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.28.4.14 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.28.4.15 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline],[inherited]`

Definition at line 208 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_hyperparams`.

8.28.4.16 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline],[inherited]`

Definition at line 160 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.28.4.17 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.28.4.18 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline],[inherited]`

Definition at line 200 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

8.28.4.19 `const CompositeDist & mappel::PointEmitterModel::get_prior () const` `[inline],[inherited]`

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.28.4.20 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static],[inherited]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.28.4.21 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()` `[static],[inherited]`

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.28.4.22 `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()`.

8.28.4.23 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const` `[inline]`,
`[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::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().

8.28.4.24 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const` `[inline]`,`[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.28.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const` `[inline]`,`[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::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().

8.28.4.26 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const`
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.28.4.27 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline]`,`[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.28.4.28 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.28.4.29 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.28.4.30 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.28.4.31 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.28.4.32 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.28.4.33 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_l, double kappa = default_intensity_kappa)` `[static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_l.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.28.4.34 **prior_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make_prior_component_position_beta (*IdxT size*, *double pos_beta = default_beta_pos*) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), and mappel::Gauss2DsModel::make_prior_beta_position().

8.28.4.35 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (*IdxT size*, *double pos_sigma = default_sigma_pos*) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.28.4.36 **prior_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make_prior_component_sigma (*double min_sigma*, *double max_sigma*, *double alpha = default_alpha_sigma*) [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.28.4.37 **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=().

8.28.4.38 **MCMCAdaptor1Ds & mappel::MCMCAdaptor1Ds::operator= (MCMCAdaptor1Ds && o)** [protected]

Definition at line 43 of file MCMCAdaptor1Ds.cpp.

References eta_sigma, and mappel::MCMCAdaptor1D::operator=().

8.28.4.39 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const`
`[inherited]`

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

8.28.4.40 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const`
`[inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.28.4.41 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)`
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.28.4.42 `void mappel::MCMCAdaptor1Ds::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)`

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`.

8.28.4.43 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.28.4.44 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ()` `[inline], [inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.28.4.45 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

8.28.4.46 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.28.4.47 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc)` [inline], [inherited]

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.28.4.48 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` [inline], [inherited]

Definition at line 240 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.28.4.49 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` [inline], [inherited]

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

8.28.4.50 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1)` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

8.28.4.51 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.28.4.52 `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()`.

8.28.4.53 `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`.

8.28.4.54 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc)` `[inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.28.4.55 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

8.28.4.56 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.28.4.57 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` `[static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.28.4.58 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

8.28.4.59 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const` `[inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.28.4.60 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const` `[inherited]`

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.28.5 Member Data Documentation

8.28.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static]`, `[inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.28.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static]`, `[inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.28.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static]`, `[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.28.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.28.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

8.28.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.28.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.28.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.28.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static],[inherited]`

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

8.28.5.10 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.28.5.11 `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.28.5.12 `double mappel::MCMCAdaptor1Ds::eta_sigma = 1` `[protected]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 27 of file `MCMCAdaptor1Ds.h`.

Referenced by `get_stats()`, `MCMCAdaptor1Ds()`, `operator=()`, and `sample_mcmc_candidate()`.

8.28.5.13 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.28.5.14 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static]`, `[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.28.5.15 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static]`, `[inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.28.5.16 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static],[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.28.5.17 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static],[inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.28.5.18 `ParamT mappel::PointEmitterModel::lbound` `[protected],[inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.28.5.19 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected],[inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

8.28.5.20 `IdxT mappel::PointEmitterModel::num_params` `[protected],[inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.28.5.21 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

8.28.5.22 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.28.5.23 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`, `[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.28.5.24 `ParamT mappel::PointEmitterModel::ubound` `[protected]`, `[inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

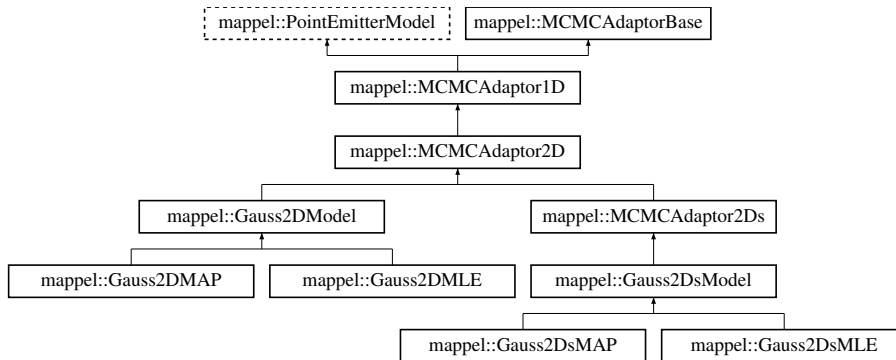
The documentation for this class was generated from the following files:

- [MCMCAdaptor1Ds.h](#)
- [MCMCAdaptor1Ds.cpp](#)

8.29 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)
- 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)`
- `ParamT sample_prior ()`
- `void set_bounds (const ParamT &lbound, const ParamT &ubound)`
- `void set_lbound (const ParamT &lbound)`
- `void set_ubound (const ParamT &ubound)`
- `const ParamT &get_lbound () const`
- `const ParamT &get_ubound () const`
- `bool theta_in_bounds (const ParamT &theta) const`
- `void bound_theta (ParamT &theta, double epsilon=bounds_epsilon) const`
- `ParamT bounded_theta (const ParamT &theta, double epsilon=bounds_epsilon) const`
- `ParamT reflected_theta (const ParamT &theta) const`
- `BoolVecT theta_stack_in_bounds (const ParamVecT &theta) const`
- `ParamVecT bounded_theta_stack (const ParamVecT &theta, double epsilon=bounds_epsilon) const`
- `ParamVecT reflected_theta_stack (const ParamVecT &theta) const`
- `void set_mcmc_sigma_scale (double scale)`
- `double get_mcmc_sigma_scale () const`
- `IdxT get_mcmc_num_phases () const`

Static Public Member Functions

- `static prior_hessian::TruncatedNormalDist make_prior_component_position_normal (IdxT size, double pos_↵, sigma=default_sigma_pos)`
- `static prior_hessian::ScaledSymmetricBetaDist make_prior_component_position_beta (IdxT size, double pos_↵, beta=default_beta_pos)`
- `static prior_hessian::TruncatedGammaDist make_prior_component_intensity (double mean=default_mean_↵, double kappa=default_intensity_kappa)`
- `static prior_hessian::TruncatedParetoDist make_prior_component_sigma (double min_sigma, double max_↵, sigma, double alpha=default_alpha_sigma)`
- `static void set_rng_seed (RngSeedT seed)`
- `static ParallelRngManagerT &get_rng_manager ()`
- `static ParallelRngGeneratorT &get_rng_generator ()`

Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5

Protected Member Functions

- [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](#)

8.29.1 Detailed Description

Definition at line 14 of file [MCMCAdaptor2D.h](#).

8.29.2 Member Typedef Documentation

8.29.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.29.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.29.3 Constructor & Destructor Documentation

8.29.3.1 `mappel::MCMCAdaptor2D::MCMCAdaptor2D ()` [protected]

Definition at line 11 of file MCMCAdaptor2D.cpp.

References `mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale`.

8.29.3.2 `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`.

8.29.3.3 `mappel::MCMCAdaptor2D::MCMCAdaptor2D (const MCMCAdaptor2D & o)` [protected]

Definition at line 22 of file MCMCAdaptor2D.cpp.

References `eta_y`.

8.29.3.4 `mappel::MCMCAdaptor2D::MCMCAdaptor2D (MCMCAdaptor2D && o)` [protected]

Definition at line 27 of file MCMCAdaptor2D.cpp.

References `eta_y`.

8.29.4 Member Function Documentation

8.29.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.29.4.2 `PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`.

8.29.4.3 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const` [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::make_param_stack()`.

8.29.4.4 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const` [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::bounded_theta_stack()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::reflected_theta_stack()`, `mappel::PointEmitterModel::theta_in_bounds()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.29.4.5 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::num_params`.

8.29.4.6 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

Referenced by `mappel::Gauss1DModel::Gauss1DModel()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss2DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::Gauss2DsModel::set_min_sigma()`, `mappel::Gauss1DModel::set_psf_sigma()`, and `mappel::Gauss2DModel::set_psf_sigma()`.

8.29.4.7 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::global_max_psf_sigma`, and `mappel::PointEmitterModel::global_min_psf_sigma`.

8.29.4.8 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.29.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`,`[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.29.4.10 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.29.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.29.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.29.4.13 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::num_phases.

8.29.4.14 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file MCMCAdaptorBase.cpp.

References mappel::MCMCAdaptorBase::sigma_scale.

8.29.4.15 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline],[inherited]`

Definition at line 208 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_hyperparams`.

8.29.4.16 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline],[inherited]`

Definition at line 160 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.29.4.17 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.29.4.18 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline],[inherited]`

Definition at line 200 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::update_internal_1Dsum_estimators()`, and `mappel::Gauss2DsModel::update_internal_1Dsum_estimators()`.

8.29.4.19 `const CompositeDist & mappel::PointEmitterModel::get_prior () const` `[inline],[inherited]`

Definition at line 204 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.29.4.20 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static],[inherited]`

Definition at line 120 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.29.4.21 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()` `[static],[inherited]`

Definition at line 115 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

8.29.4.22 `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()`.

8.29.4.23 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const` `[inline]`,
`[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_←mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.29.4.24 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const` `[inline]`,`[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.29.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const` `[inline]`,`[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::←Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel←::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_←update().

8.29.4.26 `template<class FillT> PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const`
`[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.29.4.27 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline]`,`[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.29.4.28 `template<class FillT> MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.29.4.29 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.29.4.30 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const`
`[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.29.4.31 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.29.4.32 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.29.4.33 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_I, double kappa = default_intensity_kappa)` `[static],[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_I.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss1DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.29.4.34 **prior_hessian::ScaledSymmetricBetaDist** mappel::PointEmitterModel::make_prior_component_position_beta (*IdxT size*, *double pos_beta = default_beta_pos*) [static],[inherited]

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss1DModel::make_prior_beta_position(), and mappel::Gauss2DsModel::make_prior_beta_position().

8.29.4.35 **prior_hessian::TruncatedNormalDist** mappel::PointEmitterModel::make_prior_component_position_normal (*IdxT size*, *double pos_sigma = default_sigma_pos*) [static],[inherited]

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss1DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_prior_normal_position(), mappel::Gauss2DModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.29.4.36 **prior_hessian::TruncatedParetoDist** mappel::PointEmitterModel::make_prior_component_sigma (*double min_sigma*, *double max_sigma*, *double alpha = default_alpha_sigma*) [static],[inherited]

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::Gauss1DsModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss1DsModel::make_prior_normal_position(), and mappel::Gauss2DsModel::make_prior_normal_position().

8.29.4.37 **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=().

8.29.4.38 **MCMCAdaptor2D & mappel::MCMCAdaptor2D::operator=** (*MCMCAdaptor2D && o*) [protected]

Definition at line 41 of file MCMCAdaptor2D.cpp.

References eta_y, and mappel::MCMCAdaptor1D::operator=().

8.29.4.39 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const`
`[inherited]`

Definition at line 276 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

8.29.4.40 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const`
`[inherited]`

Definition at line 317 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.29.4.41 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)`
`[inline], [inherited]`

Definition at line 244 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.29.4.42 `void mappel::MCMCAdaptor2D::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)`

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`.

8.29.4.43 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.29.4.44 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ()` `[inline], [inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.29.4.45 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` [inherited]

Definition at line 81 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_pixel_mean_bg`, `mappel::MCMCAdaptor1D::eta_bg`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_ubound()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

8.29.4.46 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)` [inherited]

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.29.4.47 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc)` [inline], [inherited]

Definition at line 260 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.29.4.48 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` [inline], [inherited]

Definition at line 240 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.29.4.49 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` [inline], [inherited]

Definition at line 220 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

8.29.4.50 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1)` [inherited]

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_I`, `mappel::MCMCAdaptor1D::eta_I`, `mappel::PointEmitterModel::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`.

8.29.4.51 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.29.4.52 `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()`.

8.29.4.53 `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`.

8.29.4.54 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc)` `[inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.29.4.55 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

8.29.4.56 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.29.4.57 void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]

Definition at line 110 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.29.4.58 void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]

Definition at line 237 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set_max_sigma(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.29.4.59 bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::make_stencil(), mappel::Gauss1DsModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2Dsxymodel::make_stencil(), mappel::Gauss2DsModel::make_stencil(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.29.4.60 BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const [inherited]

Definition at line 297 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

8.29.5 Member Data Documentation

8.29.5.1 const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6 [static],[inherited]

Distance from the boundary to constrain in bound_theta and bounded_theta methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss2DsModel::set_max_sigma_ratio(), and mappel::PointEmitterModel::set_ubound().

8.29.5.2 const double mappel::PointEmitterModel::default_alpha_sigma = 2 [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.29.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.29.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static],[inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.29.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static],[inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

8.29.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.29.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static],[inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.29.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static],[inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.29.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion" [static], [inherited]`

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

8.29.5.10 `double mappel::MCMCAdaptor1D::eta_bg=0 [protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.29.5.11 `double mappel::MCMCAdaptor1D::eta_l=0 [protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.29.5.12 `double mappel::MCMCAdaptor1D::eta_x=0 [protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.29.5.13 `double mappel::MCMCAdaptor2D::eta_y=0 [protected]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 27 of file MCMCAdaptor2D.h.

Referenced by `get_stats()`, `MCMCAdaptor2D()`, `operator=()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, and `sample_mcmc_candidate()`.

8.29.5.14 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static], [inherited]`

Definition at line 16 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `MCMCAdaptor2D()`, and `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`.

8.29.5.15 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static], [inherited]`

Definition at line 17 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.29.5.16 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.29.5.17 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.29.5.18 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.29.5.19 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

8.29.5.20 `IdxT mappel::PointEmitterModel::num_params` `[protected]`, `[inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.29.5.21 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected]`, `[inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor2Ds::sample_mcmc_candidate()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

8.29.5.22 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`, `[inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.29.5.23 `double mappel::MCMCAdaptorBase::sigma_scale` `[protected]`, `[inherited]`

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_sigma_scale()`, `mappel::MCMCAdaptorBase::get_stats()`, `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `MCMCAdaptor2D()`, `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds()`, `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`, `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.29.5.24 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::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().

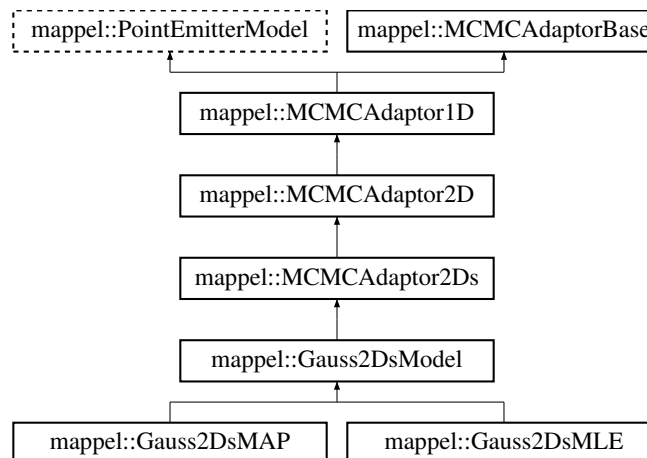
The documentation for this class was generated from the following files:

- [MCMCAdaptor2D.h](#)
- [MCMCAdaptor2D.cpp](#)

8.30 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)
- 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)
- ParamT [sample_prior](#) ()
- void [set_bounds](#) (const ParamT &lbound, const ParamT &ubound)
- void [set_lbound](#) (const ParamT &lbound)
- void [set_ubound](#) (const ParamT &ubound)
- const ParamT & [get_lbound](#) () const
- const ParamT & [get_ubound](#) () const
- bool [theta_in_bounds](#) (const ParamT &theta) const
- void [bound_theta](#) (ParamT &theta, double epsilon=[bounds_epsilon](#)) const
- ParamT [bounded_theta](#) (const ParamT &theta, double epsilon=[bounds_epsilon](#)) const
- ParamT [reflected_theta](#) (const ParamT &theta) const
- BoolVecT [theta_stack_in_bounds](#) (const ParamVecT &theta) const

- [ParamVecT bounded_theta_stack](#) (const [ParamVecT](#) &theta, double epsilon=[bounds_epsilon](#)) const
- [ParamVecT reflected_theta_stack](#) (const [ParamVecT](#) &theta) const
- void [set_mcmc_sigma_scale](#) (double scale)
- double [get_mcmc_sigma_scale](#) () const
- [IdxT get_mcmc_num_phases](#) () const

Static Public Member Functions

- static [prior_hessian::TruncatedNormalDist](#) [make_prior_component_position_normal](#) ([IdxT](#) size, double pos_↵ sigma=[default_sigma_pos](#))
- static [prior_hessian::ScaledSymmetricBetaDist](#) [make_prior_component_position_beta](#) ([IdxT](#) size, double pos_↵ beta=[default_beta_pos](#))
- static [prior_hessian::TruncatedGammaDist](#) [make_prior_component_intensity](#) (double mean=[default_mean_↵ l](#), double kappa=[default_intensity_kappa](#))
- static [prior_hessian::TruncatedParetoDist](#) [make_prior_component_sigma](#) (double min_sigma, double max_↵ sigma, double alpha=[default_alpha_sigma](#))
- static void [set_rng_seed](#) ([RngSeedT](#) seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()

Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const double [global_default_mcmc_sigma_scale](#) = 0.05
- static const double [global_max_mcmc_sigma_scale](#) = 0.5

Protected Member Functions

- [MCMCAdaptor2Ds](#) ()
- [MCMCAdaptor2Ds](#) (double [sigma_scale](#))
- [MCMCAdaptor2Ds](#) (const [MCMCAdaptor2Ds](#) &o)
- [MCMCAdaptor2Ds](#) ([MCMCAdaptor2Ds](#) &&o)
- [MCMCAdaptor2Ds](#) & [operator=](#) (const [MCMCAdaptor2Ds](#) &o)
- [MCMCAdaptor2Ds](#) & [operator=](#) ([MCMCAdaptor2Ds](#) &&o)
- [StatsT get_stats](#) () const
- void [set_mcmc_num_phases](#) ([IdxT](#) num_phases)

Protected Attributes

- double [eta_sigma](#) =0
- double [eta_y](#) =0
- double [eta_x](#) =0
- double [eta_l](#) =0
- double [eta_bg](#) =0
- CompositeDist [prior](#)
- [IdxT](#) num_params
- [IdxT](#) num_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [IdxT](#) num_phases
- double [sigma_scale](#)

8.30.1 Detailed Description

Definition at line 14 of file MCMCAdaptor2Ds.h.

8.30.2 Member Typedef Documentation

8.30.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` [inherited]

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.30.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` [inherited]

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.30.3 Constructor & Destructor Documentation

8.30.3.1 `mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds ()` [protected]

Definition at line 11 of file MCMCAdaptor2Ds.cpp.

References `mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale`.

8.30.3.2 `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`.

8.30.3.3 mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds (const MCMCAdaptor2Ds & o) [protected]

Definition at line 22 of file MCMCAdaptor2Ds.cpp.

References eta_sigma.

8.30.3.4 mappel::MCMCAdaptor2Ds::MCMCAdaptor2Ds (MCMCAdaptor2Ds && o) [protected]

Definition at line 27 of file MCMCAdaptor2Ds.cpp.

References eta_sigma.

8.30.4 Member Function Documentation

8.30.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 248 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

8.30.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded_theta_stack().

8.30.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & theta, double epsilon = bounds_epsilon) const [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::make_param_stack().

8.30.4.4 void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.30.4.5 `void mappel::PointEmitterModel::check_param_shape (const ParamVecT & theta) const` [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.30.4.6 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const` [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.30.4.7 `void mappel::PointEmitterModel::check_psf_sigma (const VecT & psf_sigma) const` [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.30.4.8 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` [inline], [inherited]

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` [inline], [inherited]

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.10 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` [inline], [inherited]

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.30.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.30.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::lbound`.

Referenced by `mappel::MCMCAdaptor1D::MCMCAdaptor1D()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.30.4.13 `IdxT mappel::MCMCAdaptorBase::get_mcmc_num_phases () const` `[inherited]`

Definition at line 56 of file `MCMCAdaptorBase.cpp`.

References `mappel::MCMCAdaptorBase::num_phases`.

8.30.4.14 `double mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const` `[inherited]`

Definition at line 53 of file `MCMCAdaptorBase.cpp`.

References `mappel::MCMCAdaptorBase::sigma_scale`.

8.30.4.15 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`, `[inherited]`

Definition at line 208 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_hyperparams`.

8.30.4.16 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`, `[inherited]`

Definition at line 160 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::num_params`.

8.30.4.17 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline]`, `[inherited]`

Definition at line 248 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.30.4.18 CompositeDist & mappel::PointEmitterModel::get_prior () [inline],[inherited]

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

8.30.4.19 const CompositeDist & mappel::PointEmitterModel::get_prior () const [inline],[inherited]

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.20 ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator () [static],[inherited]

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.30.4.21 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static],[inherited]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.30.4.22 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().

8.30.4.23 const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const [inline],[inherited]

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss2DsxModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.30.4.24 bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const [inline],[inherited]

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.25 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const [inline],[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.30.4.26 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const [inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.30.4.27 `MatT mappel::PointEmitterModel::make_param_mat () const [inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.30.4.28 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const [inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.30.4.29 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const [inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.30.4.30 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const [inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.30.4.31 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const [inline],[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bounded_theta_stack(), and mappel::PointEmitterModel::reflected_theta_stack().

8.30.4.32 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.30.4.33 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_I, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.30.4.34 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` `[static]`, `[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

8.30.4.35 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal (IdxT size, double pos_sigma = default_sigma_pos)` `[static]`, `[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.30.4.36 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma (double min_sigma, double max_sigma, double alpha = default_alpha_sigma)` `[static]`, `[inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.30.4.37 **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=().

8.30.4.38 **MCMCAdaptor2Ds & mappel::MCMCAdaptor2Ds::operator= (MCMCAdaptor2Ds && o)** [protected]

Definition at line 40 of file MCMCAdaptor2Ds.cpp.

References eta_sigma, and mappel::MCMCAdaptor2D::operator=().

8.30.4.39 **PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const** [inherited]

Definition at line 276 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::reflected_theta_stack().

8.30.4.40 **PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const** [inherited]

Definition at line 317 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::make_param_stack(), and mappel::PointEmitterModel::reflected_theta().

8.30.4.41 **void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name)** [inline],[inherited]

Definition at line 244 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.42 **void mappel::MCMCAdaptor2Ds::sample_mcmc_candidate (IdxT sample_index, ParamT & candidate, double step_scale = 1.0)**

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.

8.30.4.43 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.44 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ()` `[inline]`, `[inherited]`

Definition at line 268 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior, and mappel::rng_manager.

8.30.4.45 `void mappel::MCMCAdaptor1D::set_background_mcmc_sampling (double eta_bg = -1)` `[inherited]`

Definition at line 81 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_pixel_mean_bg, mappel::MCMCAdaptor1D::eta_bg, mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_lbound(), mappel::PointEmitterModel::get_ubound(), and mappel::MCMCAdaptorBase::sigma_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

8.30.4.46 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

8.30.4.47 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc)` `[inline]`,
`[inherited]`

Definition at line 260 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.48 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` `[inline]`,
`[inherited]`

Definition at line 240 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.49 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams) [inline],[inherited]`

Definition at line 220 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::set_hyperparams(), and mappel::Gauss2DsModel::set_hyperparams().

8.30.4.50 `void mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling (double eta_I = -1) [inherited]`

Definition at line 65 of file MCMCAdaptor1D.cpp.

References mappel::PointEmitterModel::default_mean_I, mappel::MCMCAdaptor1D::eta_I, mappel::PointEmitterModel::get_hyperparam_value(), and mappel::MCMCAdaptorBase::sigma_scale.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D().

8.30.4.51 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound) [inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounds_epsilon, mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DsModel::set_min_sigma().

8.30.4.52 `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().

8.30.4.53 `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.

8.30.4.54 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc) [inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.30.4.55 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_) [inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

8.30.4.56 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_) [inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.30.4.57 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.30.4.58 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

8.30.4.59 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const [inherited]`

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil_xy()`, `mappel::Gauss2DsxyModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.30.4.60 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const [inherited]`

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.30.5 Member Data Documentation

8.30.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` `[static], [inherited]`

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.30.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` `[static], [inherited]`

Default per-pixel background gamma distribution shape

Definition at line 62 of file `PointEmitterModel.h`.

8.30.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 56 of file `PointEmitterModel.h`.

8.30.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` `[static], [inherited]`

Default shape for intensity gamma distributions

Definition at line 60 of file `PointEmitterModel.h`.

8.30.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

8.30.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file `PointEmitterModel.h`.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.30.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.30.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.30.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

8.30.5.10 `double mappel::MCMCAdaptor1D::eta_bg = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 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_background_mcmc_sampling()`.

8.30.5.11 `double mappel::MCMCAdaptor1D::eta_l = 0` `[protected], [inherited]`

The standard deviation for the normally distributed perturbation to `theta_l` in the random walk MCMC sampling

Definition at line 31 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()`.

8.30.5.12 `double mappel::MCMCAdaptor2Ds::eta_sigma = 0` `[protected]`

The standard deviation for the normally distributed perturbation to `theta_bg` in the random walk MCMC sampling

Definition at line 26 of file `MCMCAdaptor2Ds.h`.

Referenced by `get_stats()`, `MCMCAdaptor2Ds()`, `operator=()`, and `sample_mcmc_candidate()`.

8.30.5.13 `double mappel::MCMCAdaptor1D::eta_x = 0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_x` in the random walk MCMC sampling

Definition at line 30 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()`.

8.30.5.14 `double mappel::MCMCAdaptor2D::eta_y = 0` `[protected]`, `[inherited]`

The standard deviation for the normally distributed perturbation to `theta_y` in the random walk MCMC sampling

Definition at line 27 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()`.

8.30.5.15 `const double mappel::MCMCAdaptorBase::global_default_mcmc_sigma_scale = 0.05` `[static]`, `[inherited]`

Definition at line 16 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds()`, `mappel::MCMCAdaptor2D::MCMCAdaptor2D()`, and `MCMCAdaptor2Ds()`.

8.30.5.16 `const double mappel::MCMCAdaptorBase::global_max_mcmc_sigma_scale = 0.5` `[static]`, `[inherited]`

Definition at line 17 of file `MCMCAdaptorBase.h`.

Referenced by `mappel::MCMCAdaptorBase::MCMCAdaptorBase()`, and `mappel::MCMCAdaptorBase::set_mcmc_sigma_scale()`.

8.30.5.17 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static]`, `[inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.30.5.18 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any psf_sigma. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.30.5.19 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.30.5.20 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

8.30.5.21 `IdxT mappel::PointEmitterModel::num_params` `[protected], [inherited]`

Definition at line 146 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.30.5.22 `IdxT mappel::MCMCAdaptorBase::num_phases` `[protected], [inherited]`

The number of different sampling phases for candidate selection MCMC. Each phase changes a different subset of variables.

Definition at line 29 of file MCMCAdaptorBase.h.

Referenced by `mappel::MCMCAdaptorBase::get_mcmc_num_phases()`, `mappel::MCMCAdaptorBase::get_stats()`, `sample_mcmc_candidate()`, `mappel::MCMCAdaptor2D::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1Ds::sample_mcmc_candidate()`, `mappel::MCMCAdaptor1D::sample_mcmc_candidate()`, and `mappel::MCMCAdaptorBase::set_mcmc_num_phases()`.

8.30.5.23 CompositeDist mappel::PointEmitterModel::prior [protected],[inherited]

Definition at line 145 of file PointEmitterModel.h.

Referenced by mappel::Gauss2DModel::debug_internal_sum_model_y(), mappel::Gauss2DsModel::debug_internal_sum_model_y(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::PointEmitterModel::get_hyperparam_index(), mappel::PointEmitterModel::get_hyperparam_names(), mappel::PointEmitterModel::get_hyperparam_value(), mappel::PointEmitterModel::get_hyperparams(), mappel::Gauss1DsModel::get_max_sigma(), mappel::Gauss1DsModel::get_min_sigma(), mappel::PointEmitterModel::get_param_names(), mappel::PointEmitterModel::get_prior(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::has_hyperparam(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::PointEmitterModel(), mappel::PointEmitterModel::rename_hyperparam(), mappel::PointEmitterModel::sample_prior(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_hyperparam_names(), mappel::PointEmitterModel::set_hyperparam_value(), mappel::PointEmitterModel::set_hyperparams(), mappel::PointEmitterModel::set_lbound(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::PointEmitterModel::set_param_names(), mappel::PointEmitterModel::set_prior(), and mappel::PointEmitterModel::set_ubound().

8.30.5.24 double mappel::MCMCAdaptorBase::sigma_scale [protected],[inherited]

A scaling factor for step sizes as a fraction of the size of the domain dimension we are walking in. (0.05 default)

Definition at line 30 of file MCMCAdaptorBase.h.

Referenced by mappel::MCMCAdaptorBase::get_mcmc_sigma_scale(), mappel::MCMCAdaptorBase::get_stats(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor1Ds::MCMCAdaptor1Ds(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), MCMCAdaptor2Ds(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling(), and mappel::MCMCAdaptorBase::set_mcmc_sigma_scale().

8.30.5.25 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

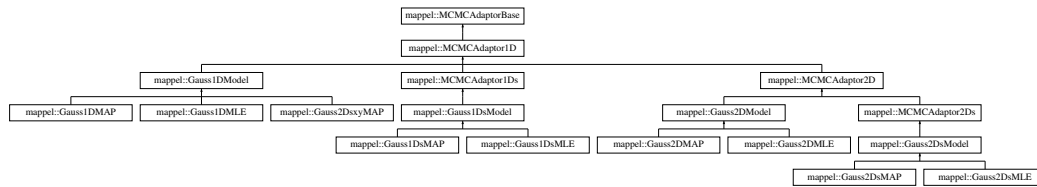
The documentation for this class was generated from the following files:

- [MCMCAdaptor2Ds.h](#)
- [MCMCAdaptor2Ds.cpp](#)

8.31 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](#)

8.31.1 Detailed Description

Definition at line 13 of file MCMCAdaptorBase.h.

8.31.2 Constructor & Destructor Documentation

8.31.2.1 mappel::MCMCAdaptorBase::MCMCAdaptorBase (IdxT num_phases) [protected]

Definition at line 14 of file MCMCAdaptorBase.cpp.

8.31.2.2 mappel::MCMCAdaptorBase::MCMCAdaptorBase (*IdxT num_phases*, *double sigma_scale*) [protected]

Definition at line 18 of file MCMCAdaptorBase.cpp.

References `global_max_mcmc_sigma_scale`.

8.31.3 Member Function Documentation

8.31.3.1 *IdxT* mappel::MCMCAdaptorBase::get_mcmc_num_phases () const

Definition at line 56 of file MCMCAdaptorBase.cpp.

References `num_phases`.

8.31.3.2 *double* mappel::MCMCAdaptorBase::get_mcmc_sigma_scale () const

Definition at line 53 of file MCMCAdaptorBase.cpp.

References `sigma_scale`.

8.31.3.3 *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()`.

8.31.3.4 *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()`.

8.31.3.5 *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`.

8.31.4 Member Data Documentation

8.31.4.1 `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()`.

8.31.4.2 `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()`.

8.31.4.3 `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()`.

8.31.4.4 `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:

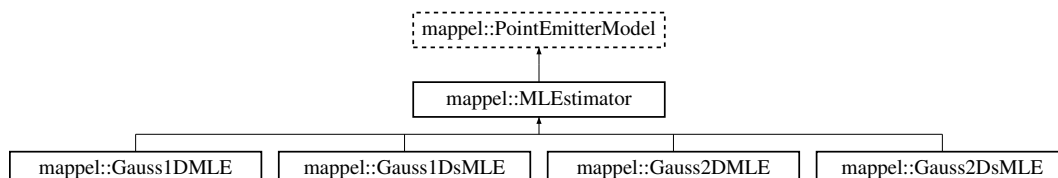
- [MCMCAdaptorBase.h](#)
- [MCMCAdaptorBase.cpp](#)

8.32 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)
- `ParamT sample_prior ()`
- void `set_bounds` (const `ParamT` &lbound, const `ParamT` &ubound)
- void `set_lbound` (const `ParamT` &lbound)
- void `set_ubound` (const `ParamT` &ubound)
- const `ParamT & get_lbound ()` const
- const `ParamT & get_ubound ()` const
- bool `theta_in_bounds` (const `ParamT` &theta) const

- void `bound_theta` (`ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `bounded_theta` (const `ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `reflected_theta` (const `ParamT` &theta) const
- `BoolVecT` `theta_stack_in_bounds` (const `ParamVecT` &theta) const
- `ParamVecT` `bounded_theta_stack` (const `ParamVecT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamVecT` `reflected_theta_stack` (const `ParamVecT` &theta) const

Static Public Member Functions

- static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` size, double pos__↔ sigma=`default_sigma_pos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos__↔ beta=`default_beta_pos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=`default_mean_l`__↔ l, double kappa=`default_intensity_kappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min_sigma, double max__↔ sigma, double alpha=`default_alpha_sigma`)
- static void `set_rng_seed` (`RngSeedT` seed)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

Static Public Attributes

- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1
- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2

Protected Member Functions

- `MLEstimator` ()=default
- `MLEstimator` (const `MLEstimator` &o)
- `MLEstimator` (`MLEstimator` &&o)
- `MLEstimator` & `operator=` (const `MLEstimator` &o)
- `MLEstimator` & `operator=` (`MLEstimator` &&o)

Protected Attributes

- `CompositeDist` `prior`
- `IdxT` `num_params`
- `IdxT` `num_hyperparams`
- `ParamT` `lbound`
- `ParamT` `ubound`

8.32.1 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.

8.32.2 Member Typedef Documentation

8.32.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec` `[inherited]`

Parameter vector

Definition at line 47 of file PointEmitterModel.h.

8.32.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat` `[inherited]`

Vector of parameter vectors

Definition at line 48 of file PointEmitterModel.h.

8.32.3 Constructor & Destructor Documentation

8.32.3.1 `mappel::MLEstimator::MLEstimator ()` `[protected]`, `[default]`

8.32.3.2 `mappel::MLEstimator::MLEstimator (const MLEstimator & o)` `[inline]`, `[protected]`

Definition at line 25 of file MLEstimator.h.

8.32.3.3 `mappel::MLEstimator::MLEstimator (MLEstimator && o)` `[inline]`, `[protected]`

Definition at line 26 of file MLEstimator.h.

8.32.4 Member Function Documentation

8.32.4.1 `void mappel::PointEmitterModel::bound_theta (ParamT & theta, double epsilon = bounds_epsilon) const`
`[inherited]`

Definition at line 248 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

8.32.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 265 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::PointEmitterModel::bounded_theta_stack().

8.32.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & *theta*, double *epsilon* = bounds_epsilon) const [inherited]

Definition at line 307 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::make_param_stack().

8.32.4.4 void mappel::PointEmitterModel::check_param_shape (const ParamT & *theta*) const [inherited]

Definition at line 167 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::bounded_theta_stack(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::reflected_theta_stack(), mappel::PointEmitterModel::theta_in_bounds(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.32.4.5 void mappel::PointEmitterModel::check_param_shape (const ParamVecT & *theta*) const [inherited]

Definition at line 176 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::num_params.

8.32.4.6 void mappel::PointEmitterModel::check_psf_sigma (double *psf_sigma*) const [inherited]

Definition at line 185 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.32.4.7 void mappel::PointEmitterModel::check_psf_sigma (const VecT & *psf_sigma*) const [inherited]

Definition at line 197 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::global_max_psf_sigma, and mappel::PointEmitterModel::global_min_psf_sigma.

8.32.4.8 `int mappel::PointEmitterModel::get_hyperparam_index (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 236 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.32.4.9 `StringVecT mappel::PointEmitterModel::get_hyperparam_names () const` `[inline]`,`[inherited]`

Definition at line 256 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.32.4.10 `double mappel::PointEmitterModel::get_hyperparam_value (const std::string & name) const` `[inline]`,
`[inherited]`

Definition at line 232 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.32.4.11 `PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const` `[inline]`,
`[inherited]`

Definition at line 224 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.32.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`,
`[inherited]`

Definition at line 212 of file PointEmitterModel.h.

References mappel::PointEmitterModel::lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.32.4.13 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`,`[inherited]`

Definition at line 208 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_hyperparams.

8.32.4.14 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`,`[inherited]`

Definition at line 160 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.32.4.15 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline],[inherited]`

Definition at line 248 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.32.4.16 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline],[inherited]`

Definition at line 200 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

8.32.4.17 `const CompositeDist & mappel::PointEmitterModel::get_prior () const` `[inline],[inherited]`

Definition at line 204 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.32.4.18 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static],[inherited]`

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.32.4.19 `ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()` `[static],[inherited]`

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.32.4.20 `StatsT mappel::PointEmitterModel::get_stats () const` `[inherited]`

Definition at line 125 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::lbound, mappel::PointEmitterModel::num_hyperparams, mappel::PointEmitterModel::num_params, mappel::PointEmitterModel::prior, mappel::rng_manager, and mappel::PointEmitterModel::ubound.

Referenced by mappel::Gauss1DModel::get_stats(), mappel::Gauss1DsModel::get_stats(), mappel::Gauss2DModel::get_stats(), and mappel::Gauss2DsModel::get_stats().

8.32.4.21 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound () const` `[inline],[inherited]`

Definition at line 216 of file PointEmitterModel.h.

References mappel::PointEmitterModel::ubound.

Referenced by mappel::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().

8.32.4.22 `bool mappel::PointEmitterModel::has_hyperparam (const std::string & name) const` `[inline],[inherited]`

Definition at line 228 of file PointEmitterModel.h.

References mappel::PointEmitterModel::prior.

8.32.4.23 `PointEmitterModel::ParamT mappel::PointEmitterModel::make_param () const` `[inline],[inherited]`

Definition at line 164 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.32.4.24 `template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill) const` `[inherited]`

Definition at line 181 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.32.4.25 `MatT mappel::PointEmitterModel::make_param_mat () const` `[inline],[inherited]`

Definition at line 172 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.32.4.26 `template<class FillT > MatT mappel::PointEmitterModel::make_param_mat (FillT fill) const` `[inherited]`

Definition at line 191 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.32.4.27 `CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const` `[inline],[inherited]`

Definition at line 176 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.32.4.28 `template<class FillT > CubeT mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 196 of file PointEmitterModel.h.

References mappel::PointEmitterModel::num_params.

8.32.4.29 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n) const` `[inline]`,
`[inherited]`

Definition at line 168 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

Referenced by `mappel::PointEmitterModel::bounded_theta_stack()`, and `mappel::PointEmitterModel::reflected_theta_stack()`.

8.32.4.30 `template<class FillT > PointEmitterModel::ParamVecT mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const` `[inherited]`

Definition at line 186 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::num_params`.

8.32.4.31 `prior_hessian::TruncatedGammaDist mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_I, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 98 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_I`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.32.4.32 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` `[static]`, `[inherited]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

8.32.4.33 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal (IdxT size, double pos_sigma = default_sigma_pos)` `[static]`, `[inherited]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.32.4.34 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma (double min_sigma, double max_sigma, double alpha = default_alpha_sigma) [static],[inherited]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.32.4.35 `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=()`.

8.32.4.36 `MLEstimator& mappel::MLEstimator::operator= (MLEstimator && o) [inline],[protected]`

Definition at line 28 of file MLEstimator.h.

8.32.4.37 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const [inherited]`

Definition at line 276 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::PointEmitterModel::reflected_theta_stack()`.

8.32.4.38 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const [inherited]`

Definition at line 317 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::make_param_stack()`, and `mappel::PointEmitterModel::reflected_theta()`.

8.32.4.39 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name) [inline],[inherited]`

Definition at line 244 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.32.4.40 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`
`[inherited]`

Definition at line 264 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.32.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior ()` `[inline]`, `[inherited]`

Definition at line 268 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`, and `mappel::rng_manager`.

8.32.4.42 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`
`[inherited]`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.32.4.43 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc)` `[inline]`,
`[inherited]`

Definition at line 260 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.32.4.44 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value)` `[inline]`,
`[inherited]`

Definition at line 240 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

8.32.4.45 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams)` `[inline]`, `[inherited]`

Definition at line 220 of file `PointEmitterModel.h`.

References `mappel::PointEmitterModel::prior`.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

8.32.4.46 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 226 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.32.4.47 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc)` `[inline],[inherited]`

Definition at line 252 of file PointEmitterModel.h.

References `mappel::PointEmitterModel::prior`.

8.32.4.48 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)` `[inherited]`

Definition at line 158 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

8.32.4.49 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)` `[inherited]`

Definition at line 149 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_hyperparams`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

8.32.4.50 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` `[static],[inherited]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.32.4.51 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 237 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::bounds_epsilon`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, `mappel::PointEmitterModel::prior`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

8.32.4.52 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const` [inherited]

Definition at line 257 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::lbound`, `mappel::PointEmitterModel::num_params`, and `mappel::PointEmitterModel::ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `mappel::PointEmitterModel::theta_stack_in_bounds()`.

8.32.4.53 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const` [inherited]

Definition at line 297 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.32.5 Member Data Documentation

8.32.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static],[inherited]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `mappel::PointEmitterModel::set_ubound()`.

8.32.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static],[inherited]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.32.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static],[inherited]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.32.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static],[inherited]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.32.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` `[static], [inherited]`

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::make_prior_component_intensity()`.

8.32.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` `[static], [inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.32.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` `[static], [inherited]`

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.32.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` `[static], [inherited]`

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.32.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` `[static], [inherited]`

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

8.32.5.10 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static], [inherited]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.32.5.11 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static], [inherited]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::check_psf_sigma()`.

8.32.5.12 `ParamT mappel::PointEmitterModel::lbound` `[protected], [inherited]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_lbound()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.32.5.13 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected], [inherited]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::get_num_hyperparams()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::operator=()`, and `mappel::PointEmitterModel::set_prior()`.

8.32.5.14 `IdxT mappel::PointEmitterModel::num_params` `[protected], [inherited]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::check_param_shape()`, `mappel::PointEmitterModel::get_num_params()`, `mappel::PointEmitterModel::get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::make_param()`, `mappel::PointEmitterModel::make_param_mat()`, `mappel::PointEmitterModel::make_param_mat_stack()`, `mappel::PointEmitterModel::make_param_stack()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.32.5.15 `CompositeDist mappel::PointEmitterModel::prior` `[protected], [inherited]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::PointEmitterModel::get_hyperparam_index()`, `mappel::PointEmitterModel::get_hyperparam_names()`, `mappel::PointEmitterModel::get_hyperparam_value()`, `mappel::PointEmitterModel::get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `mappel::PointEmitterModel::get_param_names()`, `mappel::PointEmitterModel::get_prior()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::has_hyperparam()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::PointEmitterModel()`, `mappel::PointEmitterModel::rename_hyperparam()`, `mappel::PointEmitterModel::sample_prior()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_hyperparam_names()`, `mappel::PointEmitterModel::set_hyperparam_value()`, `mappel::PointEmitterModel::set_hyperparams()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `mappel::PointEmitterModel::set_param_names()`, `mappel::PointEmitterModel::set_prior()`, and `mappel::PointEmitterModel::set_ubound()`.

8.32.5.16 ParamT mappel::PointEmitterModel::ubound [protected],[inherited]

Definition at line 148 of file PointEmitterModel.h.

Referenced by mappel::PointEmitterModel::bound_theta(), mappel::PointEmitterModel::bounded_theta(), mappel::PointEmitterModel::get_stats(), mappel::PointEmitterModel::get_ubound(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

The documentation for this class was generated from the following file:

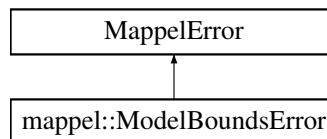
- [MLEstimator.h](#)

8.33 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)

8.33.1 Detailed Description

Access outside the model bounds is attempted.

Definition at line 86 of file util.h.

8.33.2 Constructor & Destructor Documentation

8.33.2.1 mappel::ModelBoundsError::ModelBoundsError (std::string *message*) [inline]

Definition at line 88 of file util.h.

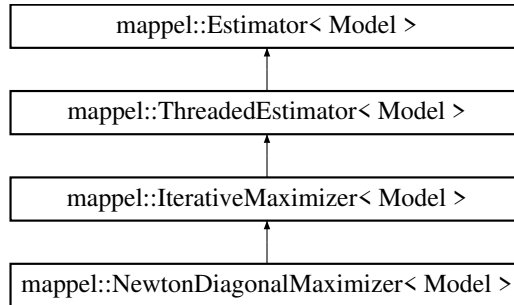
The documentation for this struct was generated from the following file:

- [util.h](#)

8.34 mappel::NewtonDiagonalMaximizer< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::NewtonDiagonalMaximizer< Model >:



Public Types

- using `MaximizerData` = typename `IterativeMaximizer< Model >::MaximizerData`
- enum `ExitCode` : `IdxT` {
`ExitCode::Unassigned` = 8, `ExitCode::MaxIter` = 7, `ExitCode::MaxBacktracks` = 6, `ExitCode::TrustRegionRadius` = 5,
`ExitCode::GradRatio` = 4, `ExitCode::FunctionChange` = 3, `ExitCode::StepSize` = 2, `ExitCode::Success` = 1,
`ExitCode::Error` = 0 }

Public Member Functions

- `NewtonDiagonalMaximizer` (`Model &model`, `int max_iterations=DEFAULT_ITERATIONS`)
- `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` ()
- `void local_maximize` (const `ModelDataT< Model > &im`, const `StencilT< Model > &theta_init`, `StencilT< Model > &stencil`, `double &rllh`)
Perform a local maximization to finish off a simulated annealing run.
- `void estimate_max_stack` (const `ModelDataStackT< Model > &im`, const `ParamVecT< Model > &theta_init`, `ParamVecT< Model > &theta`, `VecT &rllh`, `CubeT &obsI`)
- `void estimate_max_stack` (const `ModelDataStackT< Model > &im_stack`, `ParamVecT< Model > &theta_est_←` stack, `VecT &rllh_stack`, `CubeT &obsI_stack`)
- `void estimate_profile_stack` (const `ModelDataT< Model > &data`, const `IdxVecT &fixed_parameters`, const `MatT &values`, const `ParamVecT< Model > &theta_init`, `VecT &profile_likelihood`, `ParamVecT< Model > &profile_←` parameters)
- `Model &get_model` ()
- `void set_model` (`Model &new_model`)

- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, double &rllh)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, double &rllh)
- void [estimate_max](#) (const [ModelDataT](#)< Model > &im, [ParamT](#)< Model > &theta, double &rllh, [MatT](#) &obsl)
- void [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta, double &rllh, [MatT](#) &obsl)
- void [estimate_max_debug](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsl, [MatT](#) &sequence, [VecT](#) &sequence_rllh)

Static Public Attributes

- static constexpr int [NumExitCodes](#) = 9

Protected Member Functions

- void [maximize](#) ([MaximizerData](#) &data)
 - void [record_run_statistics](#) (const [MaximizerData](#) &data)
 - [StencilT](#)< Model > [compute_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, double &rllh)
 - virtual void [compute_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsl)
- Default base class implementation computes rllh and obsl separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.*
- [StencilT](#)< Model > [compute_estimate_debug](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamVecT](#)< Model > &sequence, [VecT](#) &sequence_rllh)
 - void [compute_profile_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, const [IdxVecT](#) &fixed_parameters, [ParamT](#)< Model > &theta_est, double &rllh)
 - virtual void [maximize](#) ([MaximizerData](#) &data)=0
 - bool [backtrack](#) ([MaximizerData](#) &data)
 - bool [convergence_test](#) ([MaximizerData](#) &data)
 - void [record_exit_code](#) ([ExitCode](#) code)
 - void [record_walltime](#) ([ClockT](#)::time_point start_walltime, int nimages)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = 1e-6
- double [delta](#) = 1e-6
- double [lambda_min](#) = 0.05
- double [alpha](#) = 1e-4
- int [max_backtracks](#) = 8
- int [total_iterations](#) = 0
- int [total_backtracks](#) = 0
- int [total_fun_evals](#) = 0
- int [total_der_evals](#) = 0
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)

- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.
- [IdxVecT](#) [exit_counts](#)

8.34.1 Detailed Description

```
template<class Model>
class mappel::NewtonDiagonalMaximizer< Model >
```

Definition at line 333 of file estimator.h.

8.34.2 Member Typedef Documentation

8.34.2.1 `template<class Model> using mappel::NewtonDiagonalMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 335 of file estimator.h.

8.34.3 Member Enumeration Documentation

8.34.3.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT [strong],[inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.34.4 Constructor & Destructor Documentation

8.34.4.1 `template<class Model> mappel::NewtonDiagonalMaximizer< Model >::NewtonDiagonalMaximizer (Model & model, int max_iterations = DEFAULT_ITERATIONS) [inline]`

Definition at line 337 of file estimator.h.

8.34.5 Member Function Documentation

8.34.5.1 `template<class Model> bool mappel::IterativeMaximizer< Model >::backtrack (MaximizerData & data)`
`[protected], [inherited]`

Definition at line 647 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::step`.

Referenced by `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.34.5.2 `template<class Model> void mappel::IterativeMaximizer< Model >::clear_stats ()` `[virtual]`,
`[inherited]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 625 of file `estimator_impl.h`.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::mtx`, `mappel::IterativeMaximizer< Model >::total_backtracks`, `mappel::IterativeMaximizer< Model >::total_der_evals`, `mappel::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::IterativeMaximizer< Model >::total_iterations`.

8.34.5.3 `template<class Model> void mappel::Estimator< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl)`
`[protected], [virtual], [inherited]`

Default base class implementation computes `rllh` and `obsl` separately from `stencil`. This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::model`, and `mappel::methods::observed_information()`.

8.34.5.4 `template<class Model> StencilT< Model > mappel::IterativeMaximizer< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh)` `[protected]`,
`[virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 753 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`.

```
8.34.5.5 template<class Model > StencilT< Model > mappel::IterativeMaximizer< Model >::compute_estimate_debug (
    const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamVecT< Model > & sequence,
    VecT & sequence_rllh ) [protected],[virtual],[inherited]
```

Estimators that produce a sequence of results (e.g. IterativeEstimators) can override this dummy debug implementation.

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 768 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh\(\)](#), [mappel::IterativeMaximizer< Model >::max_backtracks](#), [mappel::IterativeMaximizer< Model >::max_iterations](#), [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

```
8.34.5.6 template<class Model > void mappel::IterativeMaximizer< Model >::compute_profile_estimate ( const
    ModelDataT< Model > & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT<
    Model > & theta_est, double & rllh ) [protected],[virtual],[inherited]
```

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 797 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

```
8.34.5.7 template<class Model > bool mappel::IterativeMaximizer< Model >::convergence_test ( MaximizerData & data )
    [protected],[inherited]
```

Definition at line 731 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters](#), [mappel::IterativeMaximizer< Model >::MaximizerData::grad](#), [mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#).

```
8.34.5.8 template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max ( const ModelDataT<
    Model > & im ) [inherited]
```

Definition at line 64 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.34.5.9 `template<class Model> StencilT< Model> mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, const ParamT< Model> & theta_init) [inherited]`

Definition at line 82 of file estimator_impl.h.

References `mappel::Estimator< Model>::estimate_max()`, and `mappel::methods::objective::rllh()`.

8.34.5.10 `template<class Model> StencilT< Model> mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, double & rllh) [inherited]`

Definition at line 73 of file estimator_impl.h.

References `mappel::Estimator< Model>::estimate_max()`, and `mappel::Estimator< Model>::model`.

8.34.5.11 `template<class Model> StencilT< Model> mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, double & rllh) [inherited]`

Definition at line 90 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::print_image()`, `mappel::Estimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::record_walltime()`.

8.34.5.12 `template<class Model> void mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, ParamT< Model> & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 119 of file estimator_impl.h.

References `mappel::Estimator< Model>::estimate_max()`, and `mappel::Estimator< Model>::model`.

8.34.5.13 `template<class Model> void mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::print_image()`, `mappel::Estimator< Model>::record_exit_code()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::rllh()`.

8.34.5.14 `template<class Model> void mappel::Estimator< Model>::estimate_max_debug (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate_debug()`, `mappel::Estimator< Model>::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::rllh()`.

Referenced by `mappel::Estimator< Model>::~~Estimator()`.

8.34.5.15 `template<class Model> void mappel::Estimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im_stack, ParamVecT< Model > & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack)`
`[inherited]`

Definition at line 170 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::estimate_max_stack()`, and `mappel::Estimator< Model >::model`.

8.34.5.16 `template<class Model> void mappel::ThreadedEstimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im, const ParamVecT< Model > & theta_init, ParamVecT< Model > & theta, VecT & rllh, CubeT & obsl)` `[virtual]`, `[inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 285 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.34.5.17 `template<class Model> void mappel::ThreadedEstimator< Model >::estimate_profile_stack (const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model > & theta_init, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters)` `[virtual]`, `[inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 323 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.34.5.18 `template<class Model> StatsT mappel::IterativeMaximizer< Model >::get_debug_stats ()` `[virtual]`, `[inherited]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 609 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::IterativeMaximizer< Model >::last_backtrack_idx`s.

8.34.5.19 `template<class Model> Model & mappel::Estimator< Model >::get_model ()` `[inherited]`

Definition at line 50 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::model`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.34.5.20 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_stats () [virtual],
[inherited]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 588 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::IterativeMaximizer< Model >::max_iterations`, `mappel::ThreadedEstimator< Model >::mtx`, `mappel::Estimator< Model >::num_estimations`, `mappel::IterativeMaximizer< Model >::total_backtracks`, `mappel::IterativeMaximizer< Model >::total_der_evals`, `mappel::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::get_debug_stats()`.

8.34.5.21 `template<class Model > void mappel::IterativeMaximizer< Model >::local_maximize (const ModelDataT< Model > & im, const StencilT< Model > & theta_init, StencilT< Model > & stencil, double & rllh)
[inherited]`

Perform a local maximization to finish off a simulated annealing run.

Definition at line 811 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`.

8.34.5.22 `template<class Model > virtual void mappel::IterativeMaximizer< Model >::maximize (MaximizerData & data)
[protected], [pure virtual], [inherited]`

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, and `mappel::IterativeMaximizer< Model >::local_maximize()`.

8.34.5.23 `template<class Model > void mappel::NewtonDiagonalMaximizer< Model >::maximize (MaximizerData & data) [protected]`

Definition at line 820 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::methods::objective::grad2()`, `mappel::IterativeMaximizer< Model >::max_iterations`, `mappel::Estimator< Model >::model`, and `mappel::ThreadedEstimator< Model >::record_exit_code()`.

8.34.5.24 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_backtracks ()` `[inherited]`

8.34.5.25 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_der_evals ()` `[inherited]`

8.34.5.26 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_fun_evals ()` `[inherited]`

8.34.5.27 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_iterations ()` `[inherited]`

8.34.5.28 `template<class Model> std::string mappel::NewtonDiagonalMaximizer< Model >::name () const`
`[inline], [virtual]`

Implements [mappel::Estimator< Model >](#).

Definition at line 340 of file estimator.h.

8.34.5.29 `template<class Model > void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code)`
`[protected], [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 386 of file estimator_impl.h.

References [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::mtx](#), and [mappel::methods::objective::rllh\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::convergence_test\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

8.34.5.30 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const`
`MaximizerData & data)` `[protected], [inherited]`

Definition at line 636 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idxes\(\)](#), [mappel::IterativeMaximizer< Model >::last_backtrack_idxes](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::IterativeMaximizer< Model >::MaximizerData::nBacktracks](#), [mappel::IterativeMaximizer< Model >::MaximizerData::nIterations](#), [mappel::IterativeMaximizer< Model >::MaximizerData::save_seq](#), [mappel::IterativeMaximizer< Model >::total_backtracks](#), [mappel::IterativeMaximizer< Model >::total_der_evals](#), [mappel::IterativeMaximizer< Model >::total_fun_evals](#), and [mappel::IterativeMaximizer< Model >::total_iterations](#).

Referenced by [mappel::IterativeMaximizer< Model >::compute_estimate\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), and [mappel::IterativeMaximizer< Model >::compute_profile_estimate\(\)](#).

8.34.5.31 `template<class Model > void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages)` [protected], [inherited]

Definition at line 266 of file estimator_impl.h.

References mappel::Estimator< Model >::num_estimations, and mappel::Estimator< Model >::total_walltime.

Referenced by mappel::Estimator< Model >::estimate_max(), mappel::Estimator< Model >::estimate_max_↵ debug(), mappel::ThreadedEstimator< Model >::estimate_max_stack(), and mappel::ThreadedEstimator< Model >::estimate_profile_stack().

8.34.5.32 `template<class Model > void mappel::Estimator< Model >::set_model (Model & new_model)` [inherited]

Definition at line 54 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::~~Estimator().

8.34.6 Member Data Documentation

8.34.6.1 `template<class Model > double mappel::IterativeMaximizer< Model >::alpha = 1e-4` [protected], [inherited]

Definition at line 245 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::backtrack(), and mappel::TrustRegionMaximizer< Model >↵ ::bound_step().

8.34.6.2 `template<class Model > double mappel::IterativeMaximizer< Model >::delta = 1e-6` [protected], [inherited]

Definition at line 242 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::convergence_test(), mappel::IterativeMaximizer< Model >::get_stats(), mappel::NewtonMaximizer< Model >::maximize(), mappel::QuasiNewtonMaximizer< Model >↵ ::maximize(), mappel::TrustRegionMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >↵ ::solve_restricted_step_length_newton(), and mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem().

8.34.6.3 `template<class Model > double mappel::IterativeMaximizer< Model >::epsilon = 1e-6` [protected], [inherited]

Definition at line 241 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::convergence_test(), mappel::IterativeMaximizer< Model >↵ ::get_stats(), mappel::TrustRegionMaximizer< Model >::maximize(), and mappel::TrustRegionMaximizer< Model >↵ ::solve_restricted_step_length_newton().

8.34.6.4 `template<class Model > IdxVecT mappel::Estimator< Model >::exit_counts` [protected],
[inherited]

Definition at line 114 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, and `mappel::ThreadedEstimator< Model >::record_exit_code()`.

8.34.6.5 `template<class Model > double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` [protected],
[inherited]

Definition at line 244 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`, and `mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem()`.

8.34.6.6 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::last_backtrack_idx`s [protected],
[inherited]

Definition at line 255 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.34.6.7 `template<class Model > int mappel::IterativeMaximizer< Model >::max_backtracks = 8` [protected],
[inherited]

Definition at line 246 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, and `mappel::IterativeMaximizer< Model >::get_stats()`.

8.34.6.8 `template<class Model > int mappel::IterativeMaximizer< Model >::max_iterations` [protected],
[inherited]

Definition at line 238 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.34.6.9 `template<class Model > int mappel::ThreadedEstimator< Model >::max_threads` [protected],
[inherited]

Definition at line 142 of file estimator.h.

8.34.6.10 `template<class Model> Model& mappel::Estimator< Model>::model` [protected], [inherited]

Definition at line 109 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model>::anneal()`, `mappel::IterativeMaximizer< Model>::backtrack()`, `mappel::Estimator< Model>::compute_estimate()`, `mappel::CGaussMLE< Model>::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate()`, `mappel::IterativeMaximizer< Model>::compute_estimate()`, `mappel::Estimator< Model>::compute_estimate_debug()`, `mappel::CGaussMLE< Model>::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::Estimator< Model>::compute_profile_estimate()`, `mappel::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::estimate_max()`, `mappel::Estimator< Model>::estimate_max_debug()`, `mappel::Estimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model>::get_debug_stats()`, `mappel::Estimator< Model>::get_model()`, `mappel::IterativeMaximizer< Model>::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model>::maximize()`, `mappel::NewtonMaximizer< Model>::maximize()`, `mappel::QuasiNewtonMaximizer< Model>::maximize()`, `mappel::TrustRegionMaximizer< Model>::maximize()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::set_model()`.

8.34.6.11 `template<class Model> std::mutex mappel::ThreadedEstimator< Model>::mtx` [protected], [inherited]

Definition at line 144 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model>::clear_stats()`, `mappel::IterativeMaximizer< Model>::clear_stats()`, `mappel::ThreadedEstimator< Model>::get_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::IterativeMaximizer< Model>::record_run_statistics()`.

8.34.6.12 `template<class Model> int mappel::Estimator< Model>::num_estimations = 0` [protected], [inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model>::clear_stats()`, `mappel::Estimator< Model>::get_stats()`, `mappel::ThreadedEstimator< Model>::get_stats()`, `mappel::IterativeMaximizer< Model>::get_stats()`, and `mappel::Estimator< Model>::record_walltime()`.

8.34.6.13 `template<class Model> int mappel::ThreadedEstimator< Model>::num_threads` [protected], [inherited]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model>::clear_stats()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model>::get_stats()`.

8.34.6.14 `template<class Model> constexpr int mappel::Estimator< Model>::NumExitCodes = 9` [static], [inherited]

Definition at line 39 of file estimator.h.

8.34.6.15 `template<class Model > int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected],
[inherited]

Definition at line 250 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.34.6.16 `template<class Model > int mappel::IterativeMaximizer< Model >::total_der_evals = 0` [protected],
[inherited]

Definition at line 252 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.34.6.17 `template<class Model > int mappel::IterativeMaximizer< Model >::total_fun_evals = 0` [protected],
[inherited]

Definition at line 251 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.34.6.18 `template<class Model > int mappel::IterativeMaximizer< Model >::total_iterations = 0` [protected],
[inherited]

Definition at line 249 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.34.6.19 `template<class Model > double mappel::Estimator< Model >::total_walltime = 0.` [protected],
[inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

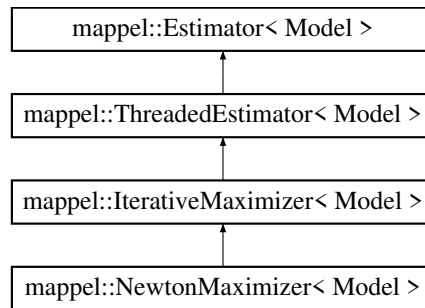
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

8.35 mappel::NewtonMaximizer< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::NewtonMaximizer< Model >:



Public Types

- using `MaximizerData` = typename `IterativeMaximizer< Model >::MaximizerData`
- enum `ExitCode` : `IdxT` {
`ExitCode::Unassigned` = 8, `ExitCode::MaxIter` = 7, `ExitCode::MaxBacktracks` = 6, `ExitCode::TrustRegionRadius` = 5,
`ExitCode::GradRatio` = 4, `ExitCode::FunctionChange` = 3, `ExitCode::StepSize` = 2, `ExitCode::Success` = 1,
`ExitCode::Error` = 0 }

Public Member Functions

- `NewtonMaximizer` (`Model &model`, `int max_iterations=DEFAULT_ITERATIONS`)
- `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` ()
- `void local_maximize` (const `ModelDataT< Model > &im`, const `StencilT< Model > &theta_init`, `StencilT< Model > &stencil`, `double &rllh`)
Perform a local maximization to finish off a simulated annealing run.
- `void estimate_max_stack` (const `ModelDataStackT< Model > &im`, const `ParamVecT< Model > &theta_init`, `ParamVecT< Model > &theta`, `VecT &rllh`, `CubeT &obsI`)
- `void estimate_max_stack` (const `ModelDataStackT< Model > &im_stack`, `ParamVecT< Model > &theta_est`, `stack`, `VecT &rllh_stack`, `CubeT &obsI_stack`)
- `void estimate_profile_stack` (const `ModelDataT< Model > &data`, const `IdxVecT &fixed_parameters`, const `MatT &values`, const `ParamVecT< Model > &theta_init`, `VecT &profile_likelihood`, `ParamVecT< Model > &profile_parameters`)
- `Model &get_model` ()
- `void set_model` (`Model &new_model`)

- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, double &rllh)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, double &rllh)
- void [estimate_max](#) (const [ModelDataT](#)< Model > &im, [ParamT](#)< Model > &theta, double &rllh, [MatT](#) &obsl)
- void [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta, double &rllh, [MatT](#) &obsl)
- void [estimate_max_debug](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsl, [MatT](#) &sequence, [VecT](#) &sequence_rllh)

Static Public Attributes

- static constexpr int [NumExitCodes](#) = 9

Protected Member Functions

- void [maximize](#) ([MaximizerData](#) &data)
 - void [record_run_statistics](#) (const [MaximizerData](#) &data)
 - [StencilT](#)< Model > [compute_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, double &rllh)
 - virtual void [compute_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamT](#)< Model > &theta_est, double &rllh, [MatT](#) &obsl)
- Default base class implementation computes rllh and obsl separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.*
- [StencilT](#)< Model > [compute_estimate_debug](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, [ParamVecT](#)< Model > &sequence, [VecT](#) &sequence_rllh)
 - void [compute_profile_estimate](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init, const [IdxVecT](#) &fixed_parameters, [ParamT](#)< Model > &theta_est, double &rllh)
 - virtual void [maximize](#) ([MaximizerData](#) &data)=0
 - bool [backtrack](#) ([MaximizerData](#) &data)
 - bool [convergence_test](#) ([MaximizerData](#) &data)
 - void [record_exit_code](#) ([ExitCode](#) code)
 - void [record_walltime](#) ([ClockT](#)::time_point start_walltime, int nimages)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = 1e-6
- double [delta](#) = 1e-6
- double [lambda_min](#) = 0.05
- double [alpha](#) = 1e-4
- int [max_backtracks](#) = 8
- int [total_iterations](#) = 0
- int [total_backtracks](#) = 0
- int [total_fun_evals](#) = 0
- int [total_der_evals](#) = 0
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)

- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.
- [IdxVecT](#) [exit_counts](#)

8.35.1 Detailed Description

```
template<class Model>
class mappel::NewtonMaximizer< Model >
```

Definition at line 347 of file estimator.h.

8.35.2 Member Typedef Documentation

8.35.2.1 `template<class Model > using mappel::NewtonMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 349 of file estimator.h.

8.35.3 Member Enumeration Documentation

8.35.3.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT [strong],[inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.35.4 Constructor & Destructor Documentation

8.35.4.1 `template<class Model > mappel::NewtonMaximizer< Model >::NewtonMaximizer (Model & model, int max_iterations = DEFAULT_ITERATIONS) [inline]`

Definition at line 351 of file estimator.h.

8.35.5 Member Function Documentation

8.35.5.1 `template<class Model> bool mappel::IterativeMaximizer< Model >::backtrack (MaximizerData & data)`
`[protected], [inherited]`

Definition at line 647 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::step`.

Referenced by `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.35.5.2 `template<class Model> void mappel::IterativeMaximizer< Model >::clear_stats ()` `[virtual]`,
`[inherited]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 625 of file `estimator_impl.h`.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::mtx`, `mappel::IterativeMaximizer< Model >::total_backtracks`, `mappel::IterativeMaximizer< Model >::total_der_evals`, `mappel::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::IterativeMaximizer< Model >::total_iterations`.

8.35.5.3 `template<class Model> void mappel::Estimator< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl)`
`[protected], [virtual], [inherited]`

Default base class implementation computes rllh and obsl separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::model`, and `mappel::methods::observed_information()`.

8.35.5.4 `template<class Model> StencilT< Model> mappel::IterativeMaximizer< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh)` `[protected]`,
`[virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 753 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`.

8.35.5.5 `template<class Model > StencilT< Model > mappel::IterativeMaximizer< Model >::compute_estimate_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamVecT< Model > & sequence, VecT & sequence_rllh) [protected],[virtual],[inherited]`

Estimators that produce a sequence of results (e.g. IterativeEstimators) can override this dummy debug implementation.

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 768 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh\(\)](#), [mappel::IterativeMaximizer< Model >::max_backtracks](#), [mappel::IterativeMaximizer< Model >::max_iterations](#), [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

8.35.5.6 `template<class Model > void mappel::IterativeMaximizer< Model >::compute_profile_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT< Model > & theta_est, double & rllh) [protected],[virtual],[inherited]`

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 797 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

8.35.5.7 `template<class Model > bool mappel::IterativeMaximizer< Model >::convergence_test (MaximizerData & data) [protected],[inherited]`

Definition at line 731 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters](#), [mappel::IterativeMaximizer< Model >::MaximizerData::grad](#), [mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#).

8.35.5.8 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im) [inherited]`

Definition at line 64 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.35.5.9 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init) [inherited]`

Definition at line 82 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::methods::objective::rllh().

8.35.5.10 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh) [inherited]`

Definition at line 73 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::Estimator< Model >::model.

8.35.5.11 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [inherited]`

Definition at line 90 of file estimator_impl.h.

References mappel::Estimator< Model >::compute_estimate(), mappel::Estimator< Model >::Error, mappel::Estimator< Model >::model, mappel::print_image(), mappel::Estimator< Model >::record_exit_code(), and mappel::Estimator< Model >::record_walltime().

8.35.5.12 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 119 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::Estimator< Model >::model.

8.35.5.13 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References mappel::Estimator< Model >::compute_estimate(), mappel::Estimator< Model >::Error, mappel::print_image(), mappel::Estimator< Model >::record_exit_code(), mappel::Estimator< Model >::record_walltime(), and mappel::methods::objective::rllh().

8.35.5.14 `template<class Model > void mappel::Estimator< Model >::estimate_max_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References mappel::Estimator< Model >::compute_estimate_debug(), mappel::Estimator< Model >::model, mappel::methods::observed_information(), mappel::Estimator< Model >::record_walltime(), and mappel::methods::objective::rllh().

Referenced by mappel::Estimator< Model >::~~Estimator().

8.35.5.15 `template<class Model> void mappel::Estimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im_stack, ParamVecT< Model> & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack)`
`[inherited]`

Definition at line 170 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::estimate_max_stack()`, and `mappel::Estimator< Model>::model`.

8.35.5.16 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im, const ParamVecT< Model> & theta_init, ParamVecT< Model> & theta, VecT & rllh, CubeT & obsl)` `[virtual]`, `[inherited]`

Implements `mappel::Estimator< Model>`.

Definition at line 285 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.35.5.17 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_profile_stack (const ModelDataT< Model> & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model> & theta_init, VecT & profile_likelihood, ParamVecT< Model> & profile_parameters)` `[virtual]`, `[inherited]`

Implements `mappel::Estimator< Model>`.

Definition at line 323 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::record_walltime()`.

8.35.5.18 `template<class Model> StatsT mappel::IterativeMaximizer< Model>::get_debug_stats ()` `[virtual]`, `[inherited]`

Reimplemented from `mappel::ThreadedEstimator< Model>`.

Definition at line 609 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model>::MaximizerData::backtrack_idx`s, `mappel::IterativeMaximizer< Model>::get_stats()`, and `mappel::IterativeMaximizer< Model>::last_backtrack_idx`s.

8.35.5.19 `template<class Model> Model & mappel::Estimator< Model>::get_model ()` `[inherited]`

Definition at line 50 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::model`.

Referenced by `mappel::Estimator< Model>::~~Estimator()`.

8.35.5.20 `template<class Model> StatsT mappel::IterativeMaximizer< Model >::get_stats ()` [virtual],
[inherited]

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 588 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::IterativeMaximizer< Model >::max_iterations`, `mappel::ThreadedEstimator< Model >::mtx`, `mappel::Estimator< Model >::num_estimations`, `mappel::IterativeMaximizer< Model >::total_backtracks`, `mappel::IterativeMaximizer< Model >::total_der_evals`, `mappel::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::get_debug_stats()`.

8.35.5.21 `template<class Model> void mappel::IterativeMaximizer< Model >::local_maximize (const ModelDataT< Model > & im, const StencilT< Model > & theta_init, StencilT< Model > & stencil, double & rllh)`
[inherited]

Perform a local maximization to finish off a simulated annealing run.

Definition at line 811 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`.

8.35.5.22 `template<class Model> virtual void mappel::IterativeMaximizer< Model >::maximize (MaximizerData & data)`
[protected], [pure virtual], [inherited]

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, and `mappel::IterativeMaximizer< Model >::local_maximize()`.

8.35.5.23 `template<class Model> void mappel::NewtonMaximizer< Model >::maximize (MaximizerData & data)`
[protected]

Definition at line 857 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::cholesky_convert_full_matrix()`, `mappel::cholesky_solve()`, `mappel::copy_Usym_mat()`, `mappel::IterativeMaximizer< Model >::delta`, `mappel::methods::objective::hessian()`, `mappel::IterativeMaximizer< Model >::max_iterations`, `mappel::Estimator< Model >::model`, `mappel::modified_cholesky()`, and `mappel::ThreadedEstimator< Model >::record_exit_code()`.

8.35.5.24 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_backtracks ()` [inherited]

8.35.5.25 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_der_evals ()` [inherited]

8.35.5.26 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_fun_evals ()` [inherited]

8.35.5.27 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_iterations ()` [inherited]

8.35.5.28 `template<class Model > std::string mappel::NewtonMaximizer< Model >::name () const` [inline],
[virtual]

Implements [mappel::Estimator< Model >](#).

Definition at line 354 of file estimator.h.

8.35.5.29 `template<class Model > void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code)`
[protected], [virtual], [inherited]

Implements [mappel::Estimator< Model >](#).

Definition at line 386 of file estimator_impl.h.

References [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::mtx](#), and [mappel::methods::objective::rllh\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::convergence_test\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

8.35.5.30 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const
MaximizerData & data)` [protected], [inherited]

Definition at line 636 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idxes\(\)](#), [mappel::IterativeMaximizer< Model >::last_backtrack_idxes](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::IterativeMaximizer< Model >::MaximizerData::nBacktracks](#), [mappel::IterativeMaximizer< Model >::MaximizerData::nIterations](#), [mappel::IterativeMaximizer< Model >::MaximizerData::save_seq](#), [mappel::IterativeMaximizer< Model >::total_backtracks](#), [mappel::IterativeMaximizer< Model >::total_der_evals](#), [mappel::IterativeMaximizer< Model >::total_fun_evals](#), and [mappel::IterativeMaximizer< Model >::total_iterations](#).

Referenced by [mappel::IterativeMaximizer< Model >::compute_estimate\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), and [mappel::IterativeMaximizer< Model >::compute_profile_estimate\(\)](#).

8.35.5.31 `template<class Model > void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages)` [protected], [inherited]

Definition at line 266 of file estimator_impl.h.

References mappel::Estimator< Model >::num_estimations, and mappel::Estimator< Model >::total_walltime.

Referenced by mappel::Estimator< Model >::estimate_max(), mappel::Estimator< Model >::estimate_max_↵ debug(), mappel::ThreadedEstimator< Model >::estimate_max_stack(), and mappel::ThreadedEstimator< Model >::estimate_profile_stack().

8.35.5.32 `template<class Model > void mappel::Estimator< Model >::set_model (Model & new_model)` [inherited]

Definition at line 54 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::~~Estimator().

8.35.6 Member Data Documentation

8.35.6.1 `template<class Model > double mappel::IterativeMaximizer< Model >::alpha = 1e-4` [protected], [inherited]

Definition at line 245 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::backtrack(), and mappel::TrustRegionMaximizer< Model >↵ ::bound_step().

8.35.6.2 `template<class Model > double mappel::IterativeMaximizer< Model >::delta = 1e-6` [protected], [inherited]

Definition at line 242 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::convergence_test(), mappel::IterativeMaximizer< Model >::get_stats(), mappel::NewtonMaximizer< Model >::maximize(), mappel::QuasiNewtonMaximizer< Model >↵ ::maximize(), mappel::TrustRegionMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >↵ ::solve_restricted_step_length_newton(), and mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem().

8.35.6.3 `template<class Model > double mappel::IterativeMaximizer< Model >::epsilon = 1e-6` [protected], [inherited]

Definition at line 241 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::convergence_test(), mappel::IterativeMaximizer< Model >↵ ::get_stats(), mappel::TrustRegionMaximizer< Model >::maximize(), and mappel::TrustRegionMaximizer< Model >↵ ::solve_restricted_step_length_newton().

8.35.6.4 `template<class Model > IdxVecT mappel::Estimator< Model >::exit_counts` [protected],
[inherited]

Definition at line 114 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, and `mappel::ThreadedEstimator< Model >::record_exit_code()`.

8.35.6.5 `template<class Model > double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` [protected],
[inherited]

Definition at line 244 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`, and `mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem()`.

8.35.6.6 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::last_backtrack_idx`s [protected],
[inherited]

Definition at line 255 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.35.6.7 `template<class Model > int mappel::IterativeMaximizer< Model >::max_backtracks = 8` [protected],
[inherited]

Definition at line 246 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, and `mappel::IterativeMaximizer< Model >::get_stats()`.

8.35.6.8 `template<class Model > int mappel::IterativeMaximizer< Model >::max_iterations` [protected],
[inherited]

Definition at line 238 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.35.6.9 `template<class Model > int mappel::ThreadedEstimator< Model >::max_threads` [protected],
[inherited]

Definition at line 142 of file estimator.h.

8.35.6.10 `template<class Model > Model& mappel::Estimator< Model >::model` [protected], [inherited]

Definition at line 109 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::Estimator< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::estimate_max()`, `mappel::Estimator< Model >::estimate_max_debug()`, `mappel::Estimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::Estimator< Model >::get_model()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::set_model()`.

8.35.6.11 `template<class Model > std::mutex mappel::ThreadedEstimator< Model >::mtx` [protected], [inherited]

Definition at line 144 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.35.6.12 `template<class Model > int mappel::Estimator< Model >::num_estimations = 0` [protected], [inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

8.35.6.13 `template<class Model > int mappel::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.35.6.14 `template<class Model > constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static], [inherited]

Definition at line 39 of file estimator.h.

8.35.6.15 `template<class Model > int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected],
[inherited]

Definition at line 250 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.35.6.16 `template<class Model > int mappel::IterativeMaximizer< Model >::total_der_evals = 0` [protected],
[inherited]

Definition at line 252 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.35.6.17 `template<class Model > int mappel::IterativeMaximizer< Model >::total_fun_evals = 0` [protected],
[inherited]

Definition at line 251 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.35.6.18 `template<class Model > int mappel::IterativeMaximizer< Model >::total_iterations = 0` [protected],
[inherited]

Definition at line 249 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.35.6.19 `template<class Model > double mappel::Estimator< Model >::total_walltime = 0.` [protected],
[inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::←
ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

The documentation for this class was generated from the following files:

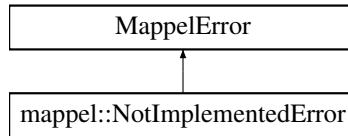
- [estimator.h](#)
- [estimator_impl.h](#)

8.36 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)

8.36.1 Detailed Description

Feature not yet implemented.

Definition at line 107 of file util.h.

8.36.2 Constructor & Destructor Documentation

8.36.2.1 mappel::NotImplementedError::NotImplementedError (std::string *message*) [inline]

Definition at line 109 of file util.h.

The documentation for this struct was generated from the following file:

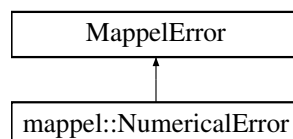
- [util.h](#)

8.37 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)

8.37.1 Detailed Description

Expected numerical condition does not hold.

Definition at line 93 of file util.h.

8.37.2 Constructor & Destructor Documentation

8.37.2.1 mappel::NumericalError::NumericalError (std::string *message*) [inline]

Definition at line 95 of file util.h.

The documentation for this struct was generated from the following file:

- [util.h](#)

8.38 omp_exception_catcher::impl::OMPExcceptionCatcher< IntType > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/OMPExcceptionCatcher/OMPExcceptionCatcher.h>
```

Public Types

- enum [Strategy](#) : IntType { [Strategy::DoNotTry](#), [Strategy::Continue](#), [Strategy::Abort](#), [Strategy::RethrowFirst](#) }

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)

8.38.1 Detailed Description

```
template<class IntType = uint32_t>
class omp_exception_catcher::impl_::OMPEXceptionCatcher< IntType >
```

Definition at line 42 of file OMPEXceptionCatcher.h.

8.38.2 Member Enumeration Documentation

```
8.38.2.1 template<class IntType = uint32_t> enum omp_exception_catcher::impl_::OMPEXceptionCatcher::Strategy
: IntType [strong]
```

Enumerator

DoNotTry
Continue
Abort
RethrowFirst

Definition at line 45 of file OMPEXceptionCatcher.h.

8.38.3 Constructor & Destructor Documentation

```
8.38.3.1 template<class IntType = uint32_t> omp_exception_catcher::impl_::OMPEXceptionCatcher< IntType
>::OMPEXceptionCatcher ( ) [inline]
```

Definition at line 50 of file OMPEXceptionCatcher.h.

```
8.38.3.2 template<class IntType = uint32_t> omp_exception_catcher::impl_::OMPEXceptionCatcher< IntType
>::OMPEXceptionCatcher ( Strategy strategy_ ) [inline]
```

Definition at line 52 of file OMPEXceptionCatcher.h.

8.38.4 Member Function Documentation

```
8.38.4.1 template<class IntType = uint32_t> void omp_exception_catcher::impl_::OMPEXceptionCatcher< IntType
>::rethrow ( ) const [inline]
```

Definition at line 54 of file OMPEXceptionCatcher.h.

References `omp_exception_catcher::impl_::OMPEXceptionCatcher< IntType >::RethrowFirst`.

Referenced by `mappel::methods::openmp::cr_lower_bound_stack()`, `mappel::methods::openmp::error_bounds↔_posterior_credible_stack()`, `mappel::methods::openmp::estimate_mcmc_posterior_stack()`, `mappel::methods↔::openmp::estimate_mcmc_sample_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()`.


```
8.38.4.2 template<class IntType = uint32_t> template<class Function , class... Parameters> void omp_exception_
_catcher::impl_::OMPEExceptionCatcher< IntType >::run ( Function func, Parameters... params )
[inline]
```

Definition at line 57 of file OMPEExceptionCatcher.h.

References `omp_exception_catcher::impl_::OMPEExceptionCatcher< IntType >::Abort`, `omp_exception_catcher::impl_::OMPEExceptionCatcher< IntType >::Continue`, `omp_exception_catcher::impl_::OMPEExceptionCatcher< IntType >::DoNotTry`, and `omp_exception_catcher::impl_::OMPEExceptionCatcher< IntType >::RethrowFirst`.

Referenced by `mappel::methods::openmp::cr_lower_bound_stack()`, `mappel::methods::openmp::error_bounds_observed_stack()`, `mappel::methods::openmp::error_bounds_posterior_credible_stack()`, `mappel::methods::openmp::estimate_mcmc_posterior_stack()`, `mappel::methods::openmp::estimate_mcmc_sample_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()`.

```
8.38.4.3 template<class IntType = uint32_t> static void omp_exception_catcher::impl_::OMPEExceptionCatcher<
IntType >::setGlobalDefaultStrategy ( Strategy s ) [inline],[static]
```

Definition at line 49 of file OMPEExceptionCatcher.h.

The documentation for this class was generated from the following file:

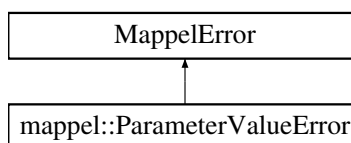
- [OMPEExceptionCatcher.h](#)

8.39 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)

8.39.1 Detailed Description

Parameter value is not valid.

Definition at line 65 of file util.h.

8.39.2 Constructor & Destructor Documentation

8.39.2.1 mappel::ParameterValueError::ParameterValueError (std::string *message*) [inline]

Definition at line 67 of file util.h.

The documentation for this struct was generated from the following file:

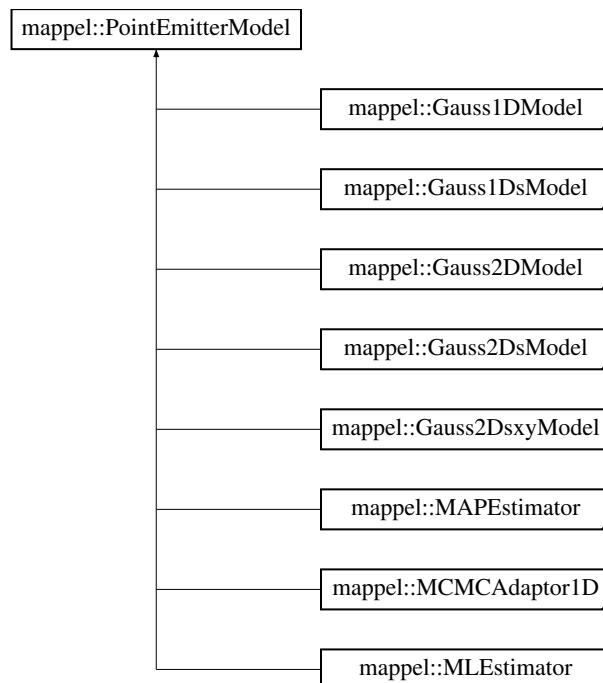
- [util.h](#)

8.40 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)
- `ParamT sample_prior ()`
- void `set_bounds` (const `ParamT` &lbound, const `ParamT` &ubound)
- void `set_lbound` (const `ParamT` &lbound)
- void `set_ubound` (const `ParamT` &ubound)
- const `ParamT` & `get_lbound ()` const
- const `ParamT` & `get_ubound ()` const
- bool `theta_in_bounds` (const `ParamT` &theta) const

- void `bound_theta` (`ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `bounded_theta` (const `ParamT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamT` `reflected_theta` (const `ParamT` &theta) const
- `BoolVecT` `theta_stack_in_bounds` (const `ParamVecT` &theta) const
- `ParamVecT` `bounded_theta_stack` (const `ParamVecT` &theta, double epsilon=`bounds_epsilon`) const
- `ParamVecT` `reflected_theta_stack` (const `ParamVecT` &theta) const

Static Public Member Functions

- static `prior_hessian::TruncatedNormalDist` `make_prior_component_position_normal` (`IdxT` size, double pos__↵ sigma=`default_sigma_pos`)
- static `prior_hessian::ScaledSymmetricBetaDist` `make_prior_component_position_beta` (`IdxT` size, double pos__↵ beta=`default_beta_pos`)
- static `prior_hessian::TruncatedGammaDist` `make_prior_component_intensity` (double mean=`default_mean_l`↵, double kappa=`default_intensity_kappa`)
- static `prior_hessian::TruncatedParetoDist` `make_prior_component_sigma` (double min_sigma, double max__↵ sigma, double alpha=`default_alpha_sigma`)
- static void `set_rng_seed` (`RngSeedT` seed)
- static `ParallelRngManagerT` & `get_rng_manager` ()
- static `ParallelRngGeneratorT` & `get_rng_generator` ()

Static Public Attributes

- static const std::string `DefaultSeperableInitEstimator` = "TrustRegion"
- static const double `bounds_epsilon` = 1.0E-6
- static const double `global_min_psf_sigma` = 1E-1
- static const double `global_max_psf_sigma` = 1E2
- static const double `default_beta_pos` = 3
- static const double `default_sigma_pos` = 1
- static const double `default_mean_l` = 300
- static const double `default_max_l` = INFINITY
- static const double `default_intensity_kappa` = 2
- static const double `default_pixel_mean_bg` = 4
- static const double `default_alpha_sigma` = 2

Protected Member Functions

- `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](#)

8.40.1 Detailed Description

A virtual Base type for point emitter localization models.

<Composite distribution from `prior_hessian::` for representing priors initialized 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`.

8.40.2 Member Typedef Documentation

8.40.2.1 `using mappel::PointEmitterModel::ParamT = arma::vec`

Parameter vector

Definition at line 47 of file `PointEmitterModel.h`.

8.40.2.2 `using mappel::PointEmitterModel::ParamVecT = arma::mat`

Vector of parameter vectors

Definition at line 48 of file `PointEmitterModel.h`.

8.40.3 Constructor & Destructor Documentation

8.40.3.1 `mappel::PointEmitterModel::PointEmitterModel ()` `[protected]`

Definition at line 31 of file `PointEmitterModel.cpp`.

8.40.3.2 `mappel::PointEmitterModel::PointEmitterModel (const CompositeDist & prior_)` `[explicit], [protected]`

Definition at line 43 of file `PointEmitterModel.cpp`.

8.40.3.3 mappel::PointEmitterModel::PointEmitterModel (CompositeDist && *prior_*) [explicit],[protected]

Definition at line 37 of file PointEmitterModel.cpp.

8.40.3.4 mappel::PointEmitterModel::PointEmitterModel (const PointEmitterModel & *o*) [protected]

Definition at line 49 of file PointEmitterModel.cpp.

References *prior*.

8.40.3.5 mappel::PointEmitterModel::PointEmitterModel (PointEmitterModel && *o*) [protected]

Definition at line 55 of file PointEmitterModel.cpp.

8.40.4 Member Function Documentation

8.40.4.1 void mappel::PointEmitterModel::bound_theta (ParamT & *theta*, double *epsilon* = bounds_epsilon) const

Definition at line 248 of file PointEmitterModel.cpp.

References *check_param_shape()*, *lbound*, *num_params*, and *ubound*.

8.40.4.2 PointEmitterModel::ParamT mappel::PointEmitterModel::bounded_theta (const ParamT & *theta*, double *epsilon* = bounds_epsilon) const

Definition at line 265 of file PointEmitterModel.cpp.

References *check_param_shape()*, *lbound*, *num_params*, and *ubound*.

Referenced by *bounded_theta_stack()*.

8.40.4.3 PointEmitterModel::ParamVecT mappel::PointEmitterModel::bounded_theta_stack (const ParamVecT & *theta*, double *epsilon* = bounds_epsilon) const

Definition at line 307 of file PointEmitterModel.cpp.

References *bounded_theta()*, *check_param_shape()*, and *make_param_stack()*.

8.40.4.4 void mappel::PointEmitterModel::check_param_shape (const ParamT & *theta*) const

Definition at line 167 of file PointEmitterModel.cpp.

References *num_params*.

Referenced by *bound_theta()*, *bounded_theta()*, *bounded_theta_stack()*, *reflected_theta()*, *reflected_theta_stack()*, *theta_in_bounds()*, and *theta_stack_in_bounds()*.

8.40.4.5 void mappel::PointEmitterModel::check_param_shape (const ParamVecT & *theta*) const

Definition at line 176 of file PointEmitterModel.cpp.

References num_params.

8.40.4.6 void mappel::PointEmitterModel::check_psf_sigma (double *psf_sigma*) const

Definition at line 185 of file PointEmitterModel.cpp.

References global_max_psf_sigma, and global_min_psf_sigma.

Referenced by mappel::Gauss1DModel::Gauss1DModel(), mappel::Gauss2DModel::Gauss2DModel(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Gauss1DsModel::set_max_sigma(), mappel::Gauss2DsModel::set_max_sigma(), mappel::Gauss1DsModel::set_min_sigma(), mappel::Gauss2DsModel::set_min_sigma(), mappel::Gauss1DModel::set_psf_sigma(), and mappel::Gauss2DModel::set_psf_sigma().

8.40.4.7 void mappel::PointEmitterModel::check_psf_sigma (const VecT & *psf_sigma*) const

Definition at line 197 of file PointEmitterModel.cpp.

References global_max_psf_sigma, and global_min_psf_sigma.

8.40.4.8 int mappel::PointEmitterModel::get_hyperparam_index (const std::string & *name*) const [inline]

Definition at line 236 of file PointEmitterModel.h.

References prior.

8.40.4.9 StringVecT mappel::PointEmitterModel::get_hyperparam_names () const [inline]

Definition at line 256 of file PointEmitterModel.h.

References prior.

8.40.4.10 double mappel::PointEmitterModel::get_hyperparam_value (const std::string & *name*) const [inline]

Definition at line 232 of file PointEmitterModel.h.

References prior.

Referenced by mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling().

8.40.4.11 PointEmitterModel::ParamT mappel::PointEmitterModel::get_hyperparams () const [inline]

Definition at line 224 of file PointEmitterModel.h.

References prior.

8.40.4.12 `const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_lbound () const` `[inline]`

Definition at line 212 of file PointEmitterModel.h.

References lbound.

Referenced by mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), and mappel::MCMCAdaptor1D::set_background_mcmc_sampling().

8.40.4.13 `IdxT mappel::PointEmitterModel::get_num_hyperparams () const` `[inline]`

Definition at line 208 of file PointEmitterModel.h.

References num_hyperparams.

8.40.4.14 `IdxT mappel::PointEmitterModel::get_num_params () const` `[inline]`

Definition at line 160 of file PointEmitterModel.h.

References num_params.

8.40.4.15 `StringVecT mappel::PointEmitterModel::get_param_names () const` `[inline]`

Definition at line 248 of file PointEmitterModel.h.

References prior.

8.40.4.16 `CompositeDist & mappel::PointEmitterModel::get_prior ()` `[inline]`

Definition at line 200 of file PointEmitterModel.h.

References prior.

Referenced by mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

8.40.4.17 `const CompositeDist & mappel::PointEmitterModel::get_prior () const` `[inline]`

Definition at line 204 of file PointEmitterModel.h.

References prior.

8.40.4.18 `ParallelRngGeneratorT & mappel::PointEmitterModel::get_rng_generator ()` `[static]`

Definition at line 120 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.40.4.19 **ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager ()** [static]

Definition at line 115 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.40.4.20 **StatsT mappel::PointEmitterModel::get_stats ()** const

Definition at line 125 of file PointEmitterModel.cpp.

References lbound, num_hyperparams, num_params, prior, mappel::rng_manager, and ubound.

Referenced by mappel::Gauss1DModel::get_stats(), mappel::Gauss1DsModel::get_stats(), mappel::Gauss2DModel::get_stats(), and mappel::Gauss2DsModel::get_stats().

8.40.4.21 **const PointEmitterModel::ParamT & mappel::PointEmitterModel::get_ubound ()** const [inline]

Definition at line 216 of file PointEmitterModel.h.

References ubound.

Referenced by mappel::Gauss2DsxyModel::get_max_sigma_ratio(), mappel::Gauss2DsModel::get_max_sigma_ratio(), mappel::MCMCAdaptor1D::MCMCAdaptor1D(), mappel::MCMCAdaptor2D::MCMCAdaptor2D(), mappel::MCMCAdaptor1D::set_background_mcmc_sampling(), and mappel::Gauss2DsModel::set_max_sigma_ratio().

8.40.4.22 **bool mappel::PointEmitterModel::has_hyperparam (const std::string & name)** const [inline]

Definition at line 228 of file PointEmitterModel.h.

References prior.

8.40.4.23 **PointEmitterModel::ParamT mappel::PointEmitterModel::make_param ()** const [inline]

Definition at line 164 of file PointEmitterModel.h.

References num_params.

Referenced by mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss1DsModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsxyModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::pixel_hess_update(), mappel::Gauss1DsModel::pixel_hess_update(), mappel::Gauss2DModel::pixel_hess_update(), and mappel::Gauss2DsModel::pixel_hess_update().

8.40.4.24 **template<class FillT > PointEmitterModel::ParamT mappel::PointEmitterModel::make_param (FillT fill)** const

Definition at line 181 of file PointEmitterModel.h.

References num_params.

8.40.4.25 **MatT** mappel::PointEmitterModel::make_param_mat () const [inline]

Definition at line 172 of file PointEmitterModel.h.

References num_params.

8.40.4.26 **template<class FillT > MatT** mappel::PointEmitterModel::make_param_mat (FillT fill) const

Definition at line 191 of file PointEmitterModel.h.

References num_params.

8.40.4.27 **CubeT** mappel::PointEmitterModel::make_param_mat_stack (IdxT n) const [inline]

Definition at line 176 of file PointEmitterModel.h.

References num_params.

8.40.4.28 **template<class FillT > CubeT** mappel::PointEmitterModel::make_param_mat_stack (IdxT n, FillT fill) const

Definition at line 196 of file PointEmitterModel.h.

References num_params.

8.40.4.29 **PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make_param_stack (IdxT n) const [inline]

Definition at line 168 of file PointEmitterModel.h.

References num_params.

Referenced by bounded_theta_stack(), and reflected_theta_stack().

8.40.4.30 **template<class FillT > PointEmitterModel::ParamVecT** mappel::PointEmitterModel::make_param_stack (IdxT n, FillT fill) const

Definition at line 186 of file PointEmitterModel.h.

References num_params.

8.40.4.31 **prior_hessian::TruncatedGammaDist** mappel::PointEmitterModel::make_prior_component_intensity (double mean = default_mean_I, double kappa = default_intensity_kappa) [static]

Definition at line 98 of file PointEmitterModel.cpp.

References default_max_I.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::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().

8.40.4.32 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos) [static]`

Definition at line 92 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2DsModel::make_prior_beta_position()`.

8.40.4.33 `prior_hessian::TruncatedNormalDist mappel::PointEmitterModel::make_prior_component_position_normal (IdxT size, double pos_sigma = default_sigma_pos) [static]`

Definition at line 85 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.40.4.34 `prior_hessian::TruncatedParetoDist mappel::PointEmitterModel::make_prior_component_sigma (double min_sigma, double max_sigma, double alpha = default_alpha_sigma) [static]`

Definition at line 104 of file PointEmitterModel.cpp.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

8.40.4.35 `PointEmitterModel & mappel::PointEmitterModel::operator= (const PointEmitterModel & o) [protected]`

Definition at line 61 of file PointEmitterModel.cpp.

References `prior`.

Referenced by `mappel::Gauss1DMap::operator=()`, `mappel::Gauss1DmLE::operator=()`, `mappel::Gauss1DsMLE::operator=()`, `mappel::Gauss1DsMap::operator=()`, `mappel::Gauss2DsMap::operator=()`, `mappel::Gauss2DsMLE::operator=()`, `mappel::Gauss2DmLE::operator=()`, and `mappel::Gauss2DMap::operator=()`.

8.40.4.36 `PointEmitterModel & mappel::PointEmitterModel::operator= (PointEmitterModel && o) [protected]`

Definition at line 68 of file PointEmitterModel.cpp.

References `lbound`, `num_hyperparams`, `num_params`, `prior`, and `ubound`.

8.40.4.37 `PointEmitterModel::ParamT mappel::PointEmitterModel::reflected_theta (const ParamT & theta) const`

Definition at line 276 of file PointEmitterModel.cpp.

References `check_param_shape()`, `lbound`, `num_params`, and `ubound`.

Referenced by `reflected_theta_stack()`.

8.40.4.38 `PointEmitterModel::ParamVecT mappel::PointEmitterModel::reflected_theta_stack (const ParamVecT & theta) const`

Definition at line 317 of file PointEmitterModel.cpp.

References `check_param_shape()`, `make_param_stack()`, and `reflected_theta()`.

8.40.4.39 `void mappel::PointEmitterModel::rename_hyperparam (const std::string & old_name, const std::string & new_name) [inline]`

Definition at line 244 of file PointEmitterModel.h.

References `prior`.

8.40.4.40 `template<class RngT > PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior (RngT & rng)`

Definition at line 264 of file PointEmitterModel.h.

References `prior`.

8.40.4.41 `PointEmitterModel::ParamT mappel::PointEmitterModel::sample_prior () [inline]`

Definition at line 268 of file PointEmitterModel.h.

References `prior`, and `mappel::rng_manager`.

8.40.4.42 `void mappel::PointEmitterModel::set_bounds (const ParamT & lbound_, const ParamT & ubound_)`

Box-type parameter bounds

Modifies the prior bounds to prevent sampling outside the valid box-constraints.

Definition at line 213 of file PointEmitterModel.cpp.

References `bounds_epsilon`, `lbound`, `num_params`, `prior`, and `ubound`.

8.40.4.43 `void mappel::PointEmitterModel::set_hyperparam_names (const StringVecT & desc) [inline]`

Definition at line 260 of file PointEmitterModel.h.

References `prior`.

8.40.4.44 `void mappel::PointEmitterModel::set_hyperparam_value (const std::string & name, double value) [inline]`

Definition at line 240 of file PointEmitterModel.h.

References prior.

8.40.4.45 `void mappel::PointEmitterModel::set_hyperparams (const VecT & hyperparams) [inline]`

Definition at line 220 of file PointEmitterModel.h.

References prior.

Referenced by `mappel::Gauss2DModel::set_hyperparams()`, and `mappel::Gauss2DsModel::set_hyperparams()`.

8.40.4.46 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)`

Definition at line 226 of file PointEmitterModel.cpp.

References `bounds_epsilon`, `lbound`, `num_params`, `prior`, and `ubound`.

Referenced by `mappel::Gauss1DsModel::set_min_sigma()`.

8.40.4.47 `void mappel::PointEmitterModel::set_param_names (const StringVecT & desc) [inline]`

Definition at line 252 of file PointEmitterModel.h.

References prior.

8.40.4.48 `void mappel::PointEmitterModel::set_prior (CompositeDist && prior_)`

Definition at line 158 of file PointEmitterModel.cpp.

References `lbound`, `num_hyperparams`, `num_params`, `prior`, and `ubound`.

Referenced by `mappel::Gauss2DModel::set_prior()`, and `mappel::Gauss2DsModel::set_prior()`.

8.40.4.49 `void mappel::PointEmitterModel::set_prior (const CompositeDist & prior_)`

Definition at line 149 of file PointEmitterModel.cpp.

References `lbound`, `num_hyperparams`, `num_params`, `prior`, and `ubound`.

8.40.4.50 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed) [static]`

Definition at line 110 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.40.4.51 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)`

Definition at line 237 of file PointEmitterModel.cpp.

References `bounds_epsilon`, `lbound`, `num_params`, `prior`, and `ubound`.

Referenced by `mappel::Gauss1DsModel::set_max_sigma()`, and `mappel::Gauss2DsModel::set_max_sigma_ratio()`.

8.40.4.52 `bool mappel::PointEmitterModel::theta_in_bounds (const ParamT & theta) const`

Definition at line 257 of file PointEmitterModel.cpp.

References `check_param_shape()`, `lbound`, `num_params`, and `ubound`.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::make_stencil()`, `mappel::Gauss1DsModel::make_stencil()`, `mappel::Gauss2DModel::make_stencil()`, `mappel::Gauss2DsxyModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil()`, and `theta_stack_in_bounds()`.

8.40.4.53 `BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & theta) const`

Definition at line 297 of file PointEmitterModel.cpp.

References `check_param_shape()`, and `theta_in_bounds()`.

8.40.5 Member Data Documentation

8.40.5.1 `const double mappel::PointEmitterModel::bounds_epsilon = 1.0E-6` [static]

Distance from the boundary to constrain in `bound_theta` and `bounded_theta` methods

Definition at line 52 of file PointEmitterModel.h.

Referenced by `set_bounds()`, `set_lbound()`, `mappel::Gauss2DsModel::set_max_sigma_ratio()`, and `set_ubound()`.

8.40.5.2 `const double mappel::PointEmitterModel::default_alpha_sigma = 2` [static]

Default per-pixel background gamma distribution shape

Definition at line 62 of file PointEmitterModel.h.

8.40.5.3 `const double mappel::PointEmitterModel::default_beta_pos = 3` [static]

Default position parameter in symmetric beta-distributions

Definition at line 56 of file PointEmitterModel.h.

8.40.5.4 `const double mappel::PointEmitterModel::default_intensity_kappa = 2` [static]

Default shape for intensity gamma distributions

Definition at line 60 of file PointEmitterModel.h.

8.40.5.5 `const double mappel::PointEmitterModel::default_max_l = INFINITY` [static]

Default maximum emitter intensity

Definition at line 59 of file PointEmitterModel.h.

Referenced by `make_prior_component_intensity()`.

8.40.5.6 `const double mappel::PointEmitterModel::default_mean_l = 300` [static]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

Referenced by `mappel::MCMCAdaptor1D::set_intensity_mcmc_sampling()`.

8.40.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` [static]

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

8.40.5.8 `const double mappel::PointEmitterModel::default_sigma_pos = 1` [static]

Default position parameter in symmetric beta-distributions

Definition at line 57 of file PointEmitterModel.h.

8.40.5.9 `const std::string mappel::PointEmitterModel::DefaultSeperableInitEstimator = "TrustRegion"` [static]

[Estimator](#) name to use in 1D separable initializations

Definition at line 49 of file PointEmitterModel.h.

Referenced by `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsxyModel::initial_theta_estimate()`, and `mappel::Gauss2DsModel::initial_theta_estimate()`.

8.40.5.10 `const double mappel::PointEmitterModel::global_max_psf_sigma = 1E2` `[static]`

Global maximum for any `psf_sigma`. Sizes above this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 54 of file `PointEmitterModel.h`.

Referenced by `check_psf_sigma()`.

8.40.5.11 `const double mappel::PointEmitterModel::global_min_psf_sigma = 1E-1` `[static]`

Global minimum for any `psf_sigma`. Sizes below this value are invalid, and nowhere near useful for practical point emitter localization

Definition at line 53 of file `PointEmitterModel.h`.

Referenced by `check_psf_sigma()`.

8.40.5.12 `ParamT mappel::PointEmitterModel::lbound` `[protected]`

Definition at line 148 of file `PointEmitterModel.h`.

Referenced by `bound_theta()`, `bounded_theta()`, `get_lbound()`, `get_stats()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `operator=()`, `reflected_theta()`, `set_bounds()`, `set_lbound()`, `set_prior()`, `set_ubound()`, and `theta_in_bounds()`.

8.40.5.13 `IdxT mappel::PointEmitterModel::num_hyperparams` `[protected]`

Definition at line 147 of file `PointEmitterModel.h`.

Referenced by `get_num_hyperparams()`, `get_stats()`, `operator=()`, and `set_prior()`.

8.40.5.14 `IdxT mappel::PointEmitterModel::num_params` `[protected]`

Definition at line 146 of file `PointEmitterModel.h`.

Referenced by `bound_theta()`, `bounded_theta()`, `check_param_shape()`, `get_num_params()`, `get_stats()`, `mappel::Gauss1DModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `make_param()`, `make_param_mat()`, `make_param_mat_stack()`, `make_param_stack()`, `operator=()`, `reflected_theta()`, `set_bounds()`, `set_lbound()`, `set_prior()`, `set_ubound()`, and `theta_in_bounds()`.

8.40.5.15 `CompositeDist mappel::PointEmitterModel::prior` `[protected]`

Definition at line 145 of file `PointEmitterModel.h`.

Referenced by `mappel::Gauss2DModel::debug_internal_sum_model_y()`, `mappel::Gauss2DsModel::debug_internal_sum_model_y()`, `mappel::Gauss2DModel::Gauss2DModel()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `get_hyperparam_index()`, `get_hyperparam_names()`, `get_hyperparam_value()`, `get_hyperparams()`, `mappel::Gauss1DsModel::get_max_sigma()`, `mappel::Gauss1DsModel::get_min_sigma()`, `get_param_names()`, `get_prior()`, `get_stats()`, `has_hyperparam()`, `operator=()`, `PointEmitterModel()`, `rename_hyperparam()`, `sample_prior()`, `set_bounds()`, `set_hyperparam_names()`, `set_hyperparam_value()`, `set_hyperparams()`, `set_lbound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::Gauss1DsModel::set_min_sigma()`, `set_param_names()`, `set_prior()`, and `set_ubound()`.

8.40.5.16 ParamT mappel::PointEmitterModel::ubound [protected]

Definition at line 148 of file PointEmitterModel.h.

Referenced by bound_theta(), bounded_theta(), get_stats(), get_ubound(), mappel::Gauss1DsModel::initial_theta_↔ estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), operator=(), reflected_theta(), set_bounds(), set_lbound(), set_prior(), set_ubound(), and theta_in_bounds().

The documentation for this class was generated from the following files:

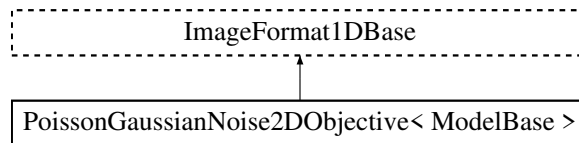
- [PointEmitterModel.h](#)
- [PointEmitterModel.cpp](#)

8.41 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](#)

8.41.1 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.

8.41.2 Member Typedef Documentation

8.41.2.1 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordIdxT = uint32_t`

Definition at line 26 of file PoissonGaussianNoise2DObjective.h.

8.41.2.2 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordStackT = arma::mat<uint32_t>`

Definition at line 28 of file PoissonGaussianNoise2DObjective.h.

8.41.2.3 `template<typename ModelBase > using PoissonGaussianNoise2DObjective< ModelBase >::CoordT = arma::vec<uint32_t>`

Definition at line 27 of file PoissonGaussianNoise2DObjective.h.

8.41.2.4 `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.

8.41.2.5 `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.

8.41.3 Constructor & Destructor Documentation

8.41.3.1 `template<typename ModelBase > PoissonGaussianNoise2DObjective< ModelBase >::PoissonGaussianNoise2DObjective (const ImageSizeVecT & size, const ImageT & sensor_gain_map, const ImageT & sensor_bg_map)`

8.41.4 Member Data Documentation

8.41.4.1 `template<typename ModelBase > const std::vector< std::string > PoissonGaussianNoise2DObjective< ModelBase >::estimator_names [static]`

Definition at line 25 of file PoissonGaussianNoise2DObjective.h.

8.41.4.2 `template<typename ModelBase > ImageT PoissonGaussianNoise2DObjective< ModelBase >::sensor_bg_map`

Definition at line 34 of file PoissonGaussianNoise2DObjective.h.

8.41.4.3 `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:

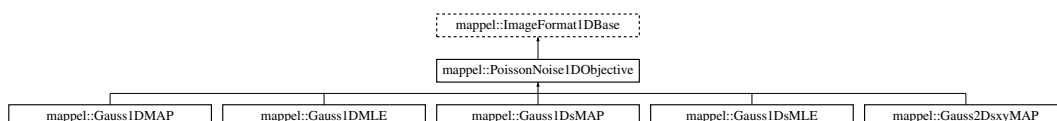
- [PoissonGaussianNoise2DObjective.h](#)
- [PoissonGaussianNoise2DObjective.cpp](#)

8.42 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

8.42.1 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`.

8.42.2 Member Typedef Documentation**8.42.2.1 using mappel::ImageFormat1DBase::ImageCoordT = uint32_t [inherited]**

Image size coordinate storage type

Definition at line 25 of file `ImageFormat1DBase.h`.

8.42.2.2 using mappel::ImageFormat1DBase::ImagePixelT = double [inherited]

Image pixel storage type

Definition at line 26 of file `ImageFormat1DBase.h`.

8.42.2.3 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`.

8.42.2.4 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`.

8.42.2.5 `using mappel::ImageFormat1DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
`[inherited]`

Data type for a single image size

Definition at line 30 of file ImageFormat1DBase.h.

8.42.2.6 `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.

8.42.2.7 `using mappel::ImageFormat1DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
`[inherited]`

Data type for a sequence of image sizes

Definition at line 31 of file ImageFormat1DBase.h.

8.42.2.8 `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.

8.42.2.9 `using mappel::ImageFormat1DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
`[inherited]`

Data type to represent a sequence of images

Definition at line 36 of file ImageFormat1DBase.h.

8.42.2.10 `using mappel::ImageFormat1DBase::ImageT = ImageShapeT<ImagePixelT>` `[inherited]`

Data type to represent single image

Definition at line 35 of file ImageFormat1DBase.h.

8.42.2.11 `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.

8.42.2.12 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.

8.42.3 Constructor & Destructor Documentation

8.42.3.1 mappel::PoissonNoise1DObjective::PoissonNoise1DObjective () [protected]

Definition at line 14 of file PoissonNoise1DObjective.cpp.

8.42.3.2 mappel::PoissonNoise1DObjective::PoissonNoise1DObjective (const PoissonNoise1DObjective & o) [protected]

Definition at line 18 of file PoissonNoise1DObjective.cpp.

8.42.3.3 mappel::PoissonNoise1DObjective::PoissonNoise1DObjective (PoissonNoise1DObjective && o) [protected]

Definition at line 22 of file PoissonNoise1DObjective.cpp.

8.42.4 Member Function Documentation

8.42.4.1 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.

8.42.4.2 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.

8.42.4.3 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().

8.42.4.4 `ImageFormat1DBase::ImageT ImageFormat1DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline],[inherited]`

Definition at line 108 of file ImageFormat1DBase.h.

8.42.4.5 `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().

8.42.4.6 `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().

8.42.4.7 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size (IdxT idx) const` `[inherited]`

Definition at line 20 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::size.

8.42.4.8 `ImageFormat1DBase::ImageCoordT ImageFormat1DBase::get_size_image_stack (const ImageStackT & stack) const` `[inline],[inherited]`

Definition at line 101 of file ImageFormat1DBase.h.

8.42.4.9 `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().

8.42.4.10 `ImageFormat1DBase::ImageT ImageFormat1DBase::make_image () const` `[inline],[inherited]`

Definition at line 87 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

8.42.4.11 `ImageFormat1DBase::ImageStackT ImageFormat1DBase::make_image_stack (ImageCoordT n) const`
`[inline], [inherited]`

Definition at line 94 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::size.

8.42.4.12 `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=().

8.42.4.13 `PoissonNoise1DObjective & mappel::PoissonNoise1DObjective::operator= (PoissonNoise1DObjective && o)`
`[protected]`

Definition at line 31 of file PoissonNoise1DObjective.cpp.

8.42.4.14 `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.

8.42.4.15 `void ImageFormat1DBase::set_size (const ImageSizeT & size_)`
`[inherited]`

Definition at line 30 of file ImageFormat1DBase.cpp.

References mappel::ImageFormat1DBase::check_size(), and mappel::ImageFormat1DBase::size.

Referenced by mappel::ImageFormat1DBase::set_size(), mappel::Gauss2DModel::set_size(), and mappel::Gauss2DsModel::set_size().

8.42.4.16 `void ImageFormat1DBase::set_size (const arma::Col< ImageCoordT > & sz)`
`[inline], [inherited]`

Definition at line 75 of file ImageFormat1DBase.h.

References mappel::ImageFormat1DBase::set_size().

8.42.5 Member Data Documentation

8.42.5.1 `const std::vector< std::string > mappel::PoissonNoise1DObjective::estimator_names`
`[static]`

Definition at line 24 of file PoissonNoise1DObjective.h.

8.42.5.2 `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()`.

8.42.5.3 `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()`.

8.42.5.4 `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()`.

8.42.5.5 `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:

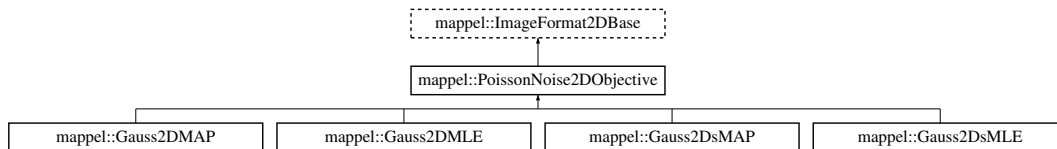
- [PoissonNoise1DObjective.h](#)
- [PoissonNoise1DObjective.cpp](#)

8.43 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](#)

8.43.1 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](#).

8.43.2 Member Typedef Documentation

8.43.2.1 using mappel::ImageFormat2DBase::ImageCoordT = uint32_t [inherited]

Image size coordinate storage type

Definition at line 24 of file [ImageFormat2DBase.h](#).

8.43.2.2 using mappel::ImageFormat2DBase::ImagePixelT = double [inherited]

Image pixel storage type

Definition at line 25 of file [ImageFormat2DBase.h](#).

8.43.2.3 `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.

8.43.2.4 `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.

8.43.2.5 `using mappel::ImageFormat2DBase::ImageSizeT = ImageSizeShapeT<ImageCoordT>`
[inherited]

Data type for a single image size

Definition at line 29 of file ImageFormat2DBase.h.

8.43.2.6 `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.

8.43.2.7 `using mappel::ImageFormat2DBase::ImageSizeVecT = ImageSizeVecShapeT<ImageCoordT>`
[inherited]

Data type for a sequence of image sizes

Definition at line 30 of file ImageFormat2DBase.h.

8.43.2.8 `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.

8.43.2.9 `using mappel::ImageFormat2DBase::ImageStackT = ImageStackShapeT<ImagePixelT>`
[inherited]

Data type to represent a sequence of images

Definition at line 35 of file ImageFormat2DBase.h.

8.43.2.10 using mappel::ImageFormat2DBase::ImageT = ImageShapeT<ImagePixelT> [inherited]

Data type to represent single image

Definition at line 34 of file ImageFormat2DBase.h.

8.43.2.11 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.

8.43.2.12 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.

8.43.3 Constructor & Destructor Documentation

8.43.3.1 mappel::PoissonNoise2DObjective::PoissonNoise2DObjective () [protected]

Definition at line 15 of file PoissonNoise2DObjective.cpp.

8.43.3.2 mappel::PoissonNoise2DObjective::PoissonNoise2DObjective (const PoissonNoise2DObjective & o) [protected]

Definition at line 19 of file PoissonNoise2DObjective.cpp.

8.43.3.3 mappel::PoissonNoise2DObjective::PoissonNoise2DObjective (PoissonNoise2DObjective && o) [protected]

Definition at line 23 of file PoissonNoise2DObjective.cpp.

8.43.4 Member Function Documentation

8.43.4.1 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.

8.43.4.2 `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.

8.43.4.3 `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().

8.43.4.4 `ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::get_image_from_stack (const ImageStackT & stack, ImageCoordT n) const` `[inline], [inherited]`

Definition at line 106 of file ImageFormat2DBase.h.

8.43.4.5 `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().

8.43.4.6 `const ImageFormat2DBase::ImageSizeT & mappel::ImageFormat2DBase::get_size () const` `[inline], [inherited]`

Definition at line 74 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.43.4.7 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size (IdxT idx) const` `[inherited]`

Definition at line 41 of file ImageFormat2DBase.cpp.

References mappel::ImageFormat2DBase::size.

8.43.4.8 `ImageFormat2DBase::ImageCoordT mappel::ImageFormat2DBase::get_size_image_stack (const ImageStackT & stack) const` `[inline], [inherited]`

Definition at line 99 of file ImageFormat2DBase.h.

8.43.4.9 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().

8.43.4.10 ImageFormat2DBase::ImageT mappel::ImageFormat2DBase::make_image () const [inline], [inherited]

Definition at line 85 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.43.4.11 ImageFormat2DBase::ImageStackT mappel::ImageFormat2DBase::make_image_stack (ImageCoordT *n*) const [inline], [inherited]

Definition at line 92 of file ImageFormat2DBase.h.

References mappel::ImageFormat2DBase::size.

8.43.4.12 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=().

8.43.4.13 PoissonNoise2DObjective & mappel::PoissonNoise2DObjective::operator= (PoissonNoise2DObjective && *o*) [protected]

Definition at line 32 of file PoissonNoise2DObjective.cpp.

8.43.4.14 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.

8.43.4.15 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().

8.43.5 Member Data Documentation

8.43.5.1 `const std::vector< std::string > mappel::PoissonNoise2DObjective::estimator_names` `[static]`

Definition at line 23 of file PoissonNoise2DObjective.h.

8.43.5.2 `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()`.

8.43.5.3 `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()`.

8.43.5.4 `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()`.

8.43.5.5 `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::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().

The documentation for this class was generated from the following files:

- [PoissonNoise2DObjective.h](#)
- [PoissonNoise2DObjective.cpp](#)

8.44 mappel::PriorMAP1DObjective Class Reference

A Mixin class to configure a [Gauss1DModel](#) for MAP estimation (default 1D prior).

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/PriorMAP1DObjective.h>
```

Protected Types

- using [ParamT](#) = arma::vec
- using [ParamMatT](#) = arma::mat

Protected Member Functions

- void [set_hyperparameters](#) (double beta_x, double mean_l, double kappa_l, double mean_bg, double kappa_bg) public
- double [prior_log_likelihood](#) (const [ParamT](#) &theta) const
- double [prior_relative_log_likelihood](#) (const [ParamT](#) &theta) const
- void [prior_grad_update](#) (const [ParamT](#) &theta, [ParamT](#) &grad) const
- void [prior_grad2_update](#) (const [ParamT](#) &theta, [ParamT](#) &grad2) const
- void [prior_hess_update](#) (const [ParamT](#) &theta, [ParamMatT](#) &hess) const

Protected Attributes

- double [log_prior_pos_const](#)
- double [log_prior_l_const](#)
- double [log_prior_bg_const](#)

8.44.1 Detailed Description

A Mixin class to configure a [Gauss1DModel](#) for MAP estimation (default 1D prior).

Definition at line 18 of file PriorMAP1DObjective.h.

8.44.2 Member Typedef Documentation

8.44.2.1 `using mappel::PriorMAP1DObjective::ParamMatT = arma::mat` `[protected]`

Definition at line 22 of file PriorMAP1DObjective.h.

8.44.2.2 `using mappel::PriorMAP1DObjective::ParamT = arma::vec` `[protected]`

Definition at line 21 of file PriorMAP1DObjective.h.

8.44.3 Member Function Documentation

8.44.3.1 `void mappel::PriorMAP1DObjective::prior_grad2_update (const ParamT & theta, ParamT & grad2) const` `[inline]`, `[protected]`

Definition at line 72 of file PriorMAP1DObjective.h.

References `mappel::beta_prior_grad2()`, `mappel::gamma_prior_grad2()`, and `mappel::methods::objective::grad2()`.

Referenced by `set_hyperparameters()`.

8.44.3.2 `void mappel::PriorMAP1DObjective::prior_grad_update (const ParamT & theta, ParamT & grad) const` `[inline]`, `[protected]`

Definition at line 64 of file PriorMAP1DObjective.h.

References `mappel::beta_prior_grad()`, `mappel::gamma_prior_grad()`, and `mappel::methods::objective::grad()`.

Referenced by `set_hyperparameters()`.

8.44.3.3 `void mappel::PriorMAP1DObjective::prior_hess_update (const ParamT & theta, ParamMatT & hess) const` `[inline]`, `[protected]`

Definition at line 80 of file PriorMAP1DObjective.h.

References `mappel::beta_prior_grad2()`, and `mappel::gamma_prior_grad2()`.

Referenced by `set_hyperparameters()`.

8.44.3.4 `double mappel::PriorMAP1DObjective::prior_log_likelihood (const ParamT & theta) const` `[inline]`,
`[protected]`

Definition at line 48 of file PriorMAP1DObjective.h.

References `log_prior_bg_const`, `log_prior_l_const`, `log_prior_pos_const`, `prior_relative_log_likelihood()`, and `mappel::methods::objective::rllh()`.

Referenced by `set_hyperparameters()`.

8.44.3.5 `double mappel::PriorMAP1DObjective::prior_relative_log_likelihood (const ParamT & theta) const` `[inline]`,
`[protected]`

Definition at line 55 of file PriorMAP1DObjective.h.

References `mappel::rllh_beta_prior()`, and `mappel::rllh_gamma_prior()`.

Referenced by `prior_log_likelihood()`, and `set_hyperparameters()`.

8.44.3.6 `void mappel::PriorMAP1DObjective::set_hyperparameters (double beta_x, double mean_l, double kappa_l, double mean_bg, double kappa_bg)` `[inline]`, `[protected]`

Definition at line 31 of file PriorMAP1DObjective.h.

References `mappel::methods::objective::grad()`, `mappel::methods::objective::grad2()`, `prior_grad2_update()`, `prior_grad_update()`, `prior_hess_update()`, `prior_log_likelihood()`, and `prior_relative_log_likelihood()`.

8.44.4 Member Data Documentation

8.44.4.1 `double mappel::PriorMAP1DObjective::log_prior_bg_const` `[protected]`

This is $\kappa_{bg} * (\log(\kappa_{bg}) - 1 / \text{mean}_{bg} - \log(\text{mean}_{bg})) - \text{lgamma}(\kappa_{bg})$

Definition at line 30 of file PriorMAP1DObjective.h.

Referenced by `prior_log_likelihood()`.

8.44.4.2 `double mappel::PriorMAP1DObjective::log_prior_l_const` `[protected]`

This is $\kappa_l * (\log(\kappa_l) - 1 / \text{mean}_l - \log(\text{mean}_l)) - \text{lgamma}(\kappa_l)$

Definition at line 29 of file PriorMAP1DObjective.h.

Referenced by `prior_log_likelihood()`.

8.44.4.3 `double mappel::PriorMAP1DObjective::log_prior_pos_const` [protected]

This is $-2 \cdot \lgamma(\text{beta_x}) - \lgamma(2 \cdot \text{beta_x})$

Definition at line 28 of file `PriorMAP1DObjective.h`.

Referenced by `prior_log_likelihood()`.

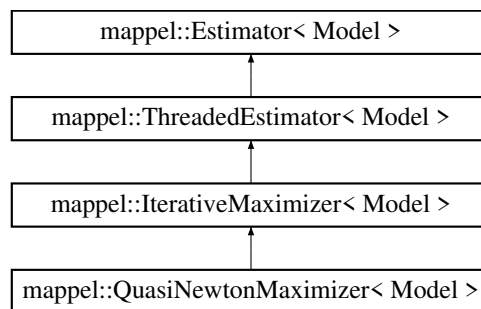
The documentation for this class was generated from the following file:

- [PriorMAP1DObjective.h](#)

8.45 `mappel::QuasiNewtonMaximizer< Model >` Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::QuasiNewtonMaximizer< Model >`:



Public Types

- using `MaximizerData` = typename `IterativeMaximizer< Model >::MaximizerData`
- enum `ExitCode` : `IdxT` {
`ExitCode::Unassigned` = 8, `ExitCode::MaxIter` = 7, `ExitCode::MaxBacktracks` = 6, `ExitCode::TrustRegionRadius` = 5,
`ExitCode::GradRatio` = 4, `ExitCode::FunctionChange` = 3, `ExitCode::StepSize` = 2, `ExitCode::Success` = 1,
`ExitCode::Error` = 0 }

Public Member Functions

- `QuasiNewtonMaximizer` (`Model &model`, int `max_iterations`=`DEFAULT_ITERATIONS`)
- `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` ()
- void `local_maximize` (const `ModelDataT`< Model > &im, const `StencilT`< Model > &theta_init, `StencilT`< Model > &stencil, double &rllh)

Perform a local maximization to finish off a simulated annealing run.
- void `estimate_max_stack` (const `ModelDataStackT`< Model > &im, const `ParamVecT`< Model > &theta_init, `ParamVecT`< Model > &theta, `VecT` &rllh, `CubeT` &obsI)
- void `estimate_max_stack` (const `ModelDataStackT`< Model > &im_stack, `ParamVecT`< Model > &theta_est, `VecT` &rllh_stack, `CubeT` &obsI_stack)
- void `estimate_profile_stack` (const `ModelDataT`< Model > &data, const `IdxVecT` &fixed_parameters, const `MatT` &values, const `ParamVecT`< Model > &theta_init, `VecT` &profile_likelihood, `ParamVecT`< Model > &profile_parameters)
- `Model` & `get_model` ()
- void `set_model` (`Model` &new_model)
- `StencilT`< Model > `estimate_max` (const `ModelDataT`< Model > &im)
- `StencilT`< Model > `estimate_max` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init)
- `StencilT`< Model > `estimate_max` (const `ModelDataT`< Model > &im, double &rllh)
- `StencilT`< Model > `estimate_max` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, double &rllh)
- void `estimate_max` (const `ModelDataT`< Model > &im, `ParamT`< Model > &theta, double &rllh, `MatT` &obsI)
- void `estimate_max` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, `ParamT`< Model > &theta, double &rllh, `MatT` &obsI)
- void `estimate_max_debug` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, `ParamT`< Model > &theta_est, double &rllh, `MatT` &obsI, `MatT` &sequence, `VecT` &sequence_rllh)

Static Public Attributes

- static constexpr int `NumExitCodes` = 9

Protected Member Functions

- void `maximize` (`MaximizerData` &data)
- void `record_run_statistics` (const `MaximizerData` &data)
- `StencilT`< Model > `compute_estimate` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, double &rllh)
- virtual void `compute_estimate` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, `ParamT`< Model > &theta_est, double &rllh, `MatT` &obsI)

Default base class implementation computes rllh and obsI separately from stencil This should be overridden by Estimator subclasses that already have access to this information.
- `StencilT`< Model > `compute_estimate_debug` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, `ParamVecT`< Model > &sequence, `VecT` &sequence_rllh)
- void `compute_profile_estimate` (const `ModelDataT`< Model > &im, const `ParamT`< Model > &theta_init, const `IdxVecT` &fixed_parameters, `ParamT`< Model > &theta_est, double &rllh)
- virtual void `maximize` (`MaximizerData` &data)=0
- bool `backtrack` (`MaximizerData` &data)
- bool `convergence_test` (`MaximizerData` &data)
- void `record_exit_code` (`ExitCode` code)
- void `record_walltime` (`ClockT`::time_point start_walltime, int nimages)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = 1e-6
- double [delta](#) = 1e-6
- double [lambda_min](#) = 0.05
- double [alpha](#) = 1e-4
- int [max_backtracks](#) = 8
- int [total_iterations](#) = 0
- int [total_backtracks](#) = 0
- int [total_fun_evals](#) = 0
- int [total_der_evals](#) = 0
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.
- [IdxVecT](#) [exit_counts](#)

8.45.1 Detailed Description

```
template<class Model>
class mappel::QuasiNewtonMaximizer< Model >
```

Definition at line 361 of file estimator.h.

8.45.2 Member Typedef Documentation

8.45.2.1 `template<class Model > using mappel::QuasiNewtonMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 363 of file estimator.h.

8.45.3 Member Enumeration Documentation

8.45.3.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT [strong], [inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.45.4 Constructor & Destructor Documentation

8.45.4.1 `template<class Model > mappel::QuasiNewtonMaximizer< Model >::QuasiNewtonMaximizer (Model & model, int max_iterations = DEFAULT_ITERATIONS) [inline]`

Definition at line 365 of file estimator.h.

8.45.5 Member Function Documentation

8.45.5.1 `template<class Model > bool mappel::IterativeMaximizer< Model >::backtrack (MaximizerData & data) [protected], [inherited]`

Definition at line 647 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, `mappel::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::step`.

Referenced by `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.45.5.2 `template<class Model > void mappel::IterativeMaximizer< Model >::clear_stats () [virtual], [inherited]`

Reimplemented from `mappel::ThreadedEstimator< Model >`.

Definition at line 625 of file estimator_impl.h.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::mtx`, `mappel::IterativeMaximizer< Model >::total_backtracks`, `mappel::IterativeMaximizer< Model >::total_der_evals`, `mappel::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::IterativeMaximizer< Model >::total_iterations`.

8.45.5.3 `template<class Model > void mappel::Estimator< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl) [protected], [virtual], [inherited]`

Default base class implementation computes rllh and obsl separately from stencil This should be overridden by `Estimator` subclasses that already have access to this information.

Definition at line 183 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::model`, and `mappel::methods::observed_information()`.

8.45.5.4 `template<class Model > StencilT< Model > mappel::IterativeMaximizer< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [protected], [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 753 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

8.45.5.5 `template<class Model > StencilT< Model > mappel::IterativeMaximizer< Model >::compute_estimate_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamVecT< Model > & sequence, VecT & sequence_rllh) [protected], [virtual], [inherited]`

Estimators that produce a sequence of results (e.g. [IterativeEstimators](#)) can override this dummy debug implementation.

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 768 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh\(\)](#), [mappel::IterativeMaximizer< Model >::max_backtracks](#), [mappel::IterativeMaximizer< Model >::max_iterations](#), [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

8.45.5.6 `template<class Model > void mappel::IterativeMaximizer< Model >::compute_profile_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT< Model > & theta_est, double & rllh) [protected], [virtual], [inherited]`

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 797 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

8.45.5.7 `template<class Model > bool mappel::IterativeMaximizer< Model >::convergence_test (MaximizerData & data) [protected], [inherited]`

Definition at line 731 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters](#), [mappel::IterativeMaximizer< Model >::MaximizerData::grad](#), [mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#).

8.45.5.8 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im) [inherited]`

Definition at line 64 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::estimate_max(), and mappel::Estimator< Model >::~Estimator().

8.45.5.9 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init) [inherited]`

Definition at line 82 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::methods::objective::rllh().

8.45.5.10 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh) [inherited]`

Definition at line 73 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::Estimator< Model >::model.

8.45.5.11 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [inherited]`

Definition at line 90 of file estimator_impl.h.

References mappel::Estimator< Model >::compute_estimate(), mappel::Estimator< Model >::Error, mappel::Estimator< Model >::model, mappel::print_image(), mappel::Estimator< Model >::record_exit_code(), and mappel::Estimator< Model >::record_walltime().

8.45.5.12 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 119 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::Estimator< Model >::model.

8.45.5.13 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References mappel::Estimator< Model >::compute_estimate(), mappel::Estimator< Model >::Error, mappel::print_image(), mappel::Estimator< Model >::record_exit_code(), mappel::Estimator< Model >::record_walltime(), and mappel::methods::objective::rllh().

8.45.5.14 `template<class Model > void mappel::Estimator< Model >::estimate_max_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.45.5.15 `template<class Model > void mappel::Estimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im_stack, ParamVecT< Model > & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack) [inherited]`

Definition at line 170 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max_stack()`, and `mappel::Estimator< Model >::model`.

8.45.5.16 `template<class Model > void mappel::ThreadedEstimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im, const ParamVecT< Model > & theta_init, ParamVecT< Model > & theta, VecT & rllh, CubeT & obsl) [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 285 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.45.5.17 `template<class Model > void mappel::ThreadedEstimator< Model >::estimate_profile_stack (const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model > & theta_init, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters) [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 323 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.45.5.18 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_debug_stats () [virtual], [inherited]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 609 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::IterativeMaximizer< Model >::last_backtrack_idx`s.

8.45.5.19 `template<class Model > Model & mappel::Estimator< Model >::get_model ()` [inherited]

Definition at line 50 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.45.5.20 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_stats ()` [virtual], [inherited]

Reimplemented from `mappel::ThreadedEstimator< Model >`.

Definition at line 588 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::IterativeMaximizer< Model >::max_iterations`, `mappel::ThreadedEstimator< Model >::mtx`, `mappel::Estimator< Model >::num_estimations`, `mappel::IterativeMaximizer< Model >::total_backtracks`, `mappel::IterativeMaximizer< Model >::total_der_evals`, `mappel::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::get_debug_stats()`.

8.45.5.21 `template<class Model > void mappel::IterativeMaximizer< Model >::local_maximize (const ModelDataT< Model > & im, const StencilT< Model > & theta_init, StencilT< Model > & stencil, double & rllh)` [inherited]

Perform a local maximization to finish off a simulated annealing run.

Definition at line 811 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`.

8.45.5.22 `template<class Model > virtual void mappel::IterativeMaximizer< Model >::maximize (MaximizerData & data)` [protected], [pure virtual], [inherited]

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, and `mappel::IterativeMaximizer< Model >::local_maximize()`.

8.45.5.23 `template<class Model > void mappel::QuasiNewtonMaximizer< Model >::maximize (MaximizerData & data)` [protected]

Definition at line 933 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::delta`, `mappel::methods::objective::grad()`, `mappel::methods::objective::hessian()`, `mappel::is_positive_definite()`, `mappel::IterativeMaximizer< Model >::max_iterations`, `mappel::Estimator< Model >::model`, and `mappel::ThreadedEstimator< Model >::record_exit_code()`.

8.45.5.24 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_backtracks () [inherited]`

8.45.5.25 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_der_evals () [inherited]`

8.45.5.26 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_fun_evals () [inherited]`

8.45.5.27 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_iterations () [inherited]`

8.45.5.28 `template<class Model > std::string mappel::QuasiNewtonMaximizer< Model >::name () const [inline], [virtual]`

Implements [mappel::Estimator< Model >](#).

Definition at line 368 of file estimator.h.

8.45.5.29 `template<class Model > void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code) [protected], [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 386 of file estimator_impl.h.

References [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::mtx](#), and [mappel::methods::objective::rllh\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::convergence_test\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

8.45.5.30 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const MaximizerData & data) [protected], [inherited]`

Definition at line 636 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idxes\(\)](#), [mappel::IterativeMaximizer< Model >::last_backtrack_idxes](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::IterativeMaximizer< Model >::MaximizerData::nBacktracks](#), [mappel::IterativeMaximizer< Model >::MaximizerData::nIterations](#), [mappel::IterativeMaximizer< Model >::MaximizerData::save_seq](#), [mappel::IterativeMaximizer< Model >::total_backtracks](#), [mappel::IterativeMaximizer< Model >::total_der_evals](#), [mappel::IterativeMaximizer< Model >::total_fun_evals](#), and [mappel::IterativeMaximizer< Model >::total_iterations](#).

Referenced by [mappel::IterativeMaximizer< Model >::compute_estimate\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), and [mappel::IterativeMaximizer< Model >::compute_profile_estimate\(\)](#).

8.45.5.31 `template<class Model > void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages)` [protected], [inherited]

Definition at line 266 of file estimator_impl.h.

References mappel::Estimator< Model >::num_estimations, and mappel::Estimator< Model >::total_walltime.

Referenced by mappel::Estimator< Model >::estimate_max(), mappel::Estimator< Model >::estimate_max_↵ debug(), mappel::ThreadedEstimator< Model >::estimate_max_stack(), and mappel::ThreadedEstimator< Model >::estimate_profile_stack().

8.45.5.32 `template<class Model > void mappel::Estimator< Model >::set_model (Model & new_model)` [inherited]

Definition at line 54 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::~~Estimator().

8.45.6 Member Data Documentation

8.45.6.1 `template<class Model > double mappel::IterativeMaximizer< Model >::alpha = 1e-4` [protected], [inherited]

Definition at line 245 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::backtrack(), and mappel::TrustRegionMaximizer< Model >↵ ::bound_step().

8.45.6.2 `template<class Model > double mappel::IterativeMaximizer< Model >::delta = 1e-6` [protected], [inherited]

Definition at line 242 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::convergence_test(), mappel::IterativeMaximizer< Model >::get_stats(), mappel::NewtonMaximizer< Model >::maximize(), mappel::QuasiNewtonMaximizer< Model >↵ ::maximize(), mappel::TrustRegionMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >↵ ::solve_restricted_step_length_newton(), and mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem().

8.45.6.3 `template<class Model > double mappel::IterativeMaximizer< Model >::epsilon = 1e-6` [protected], [inherited]

Definition at line 241 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::convergence_test(), mappel::IterativeMaximizer< Model >↵ ::get_stats(), mappel::TrustRegionMaximizer< Model >::maximize(), and mappel::TrustRegionMaximizer< Model >↵ ::solve_restricted_step_length_newton().

8.45.6.4 `template<class Model > IdxVecT mappel::Estimator< Model >::exit_counts` [protected],
[inherited]

Definition at line 114 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, and `mappel::ThreadedEstimator< Model >::record_exit_code()`.

8.45.6.5 `template<class Model > double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` [protected],
[inherited]

Definition at line 244 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`, and `mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem()`.

8.45.6.6 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::last_backtrack_idx`s [protected],
[inherited]

Definition at line 255 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.45.6.7 `template<class Model > int mappel::IterativeMaximizer< Model >::max_backtracks = 8` [protected],
[inherited]

Definition at line 246 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, and `mappel::IterativeMaximizer< Model >::get_stats()`.

8.45.6.8 `template<class Model > int mappel::IterativeMaximizer< Model >::max_iterations` [protected],
[inherited]

Definition at line 238 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.45.6.9 `template<class Model > int mappel::ThreadedEstimator< Model >::max_threads` [protected],
[inherited]

Definition at line 142 of file estimator.h.

8.45.6.10 `template<class Model > Model& mappel::Estimator< Model >::model` [protected], [inherited]

Definition at line 109 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::Estimator< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::estimate_max()`, `mappel::Estimator< Model >::estimate_max_debug()`, `mappel::Estimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::Estimator< Model >::get_model()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::set_model()`.

8.45.6.11 `template<class Model > std::mutex mappel::ThreadedEstimator< Model >::mtx` [protected], [inherited]

Definition at line 144 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.45.6.12 `template<class Model > int mappel::Estimator< Model >::num_estimations = 0` [protected], [inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

8.45.6.13 `template<class Model > int mappel::ThreadedEstimator< Model >::num_threads` [protected], [inherited]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.45.6.14 `template<class Model > constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static], [inherited]

Definition at line 39 of file estimator.h.

8.45.6.15 `template<class Model > int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected],
[inherited]

Definition at line 250 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.45.6.16 `template<class Model > int mappel::IterativeMaximizer< Model >::total_der_evals = 0` [protected],
[inherited]

Definition at line 252 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.45.6.17 `template<class Model > int mappel::IterativeMaximizer< Model >::total_fun_evals = 0` [protected],
[inherited]

Definition at line 251 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.45.6.18 `template<class Model > int mappel::IterativeMaximizer< Model >::total_iterations = 0` [protected],
[inherited]

Definition at line 249 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.45.6.19 `template<class Model > double mappel::Estimator< Model >::total_walltime = 0.` [protected],
[inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::←
ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

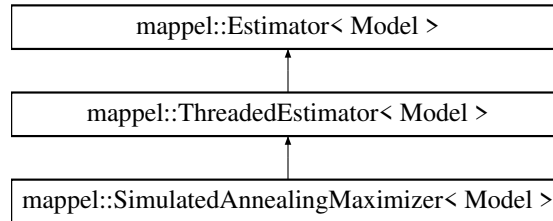
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

8.46 mappel::SimulatedAnnealingMaximizer< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::SimulatedAnnealingMaximizer< Model >:



Public Types

- enum `ExitCode` : `IdxT` {
`ExitCode::Unassigned` = 8, `ExitCode::MaxIter` = 7, `ExitCode::MaxBacktracks` = 6, `ExitCode::TrustRegionRadius` = 5,
`ExitCode::GradRatio` = 4, `ExitCode::FunctionChange` = 3, `ExitCode::StepSize` = 2, `ExitCode::Success` = 1,
`ExitCode::Error` = 0 }

Public Member Functions

- `SimulatedAnnealingMaximizer` (`Model` &`model`)
- `StatsT` `get_stats` ()
- `StatsT` `get_debug_stats` ()
- `std::string` `name` () const
- void `estimate_max_stack` (const `ModelDataStackT`< `Model` > &`im`, const `ParamVecT`< `Model` > &`theta_init`, `ParamVecT`< `Model` > &`theta`, `VecT` &`rllh`, `CubeT` &`obsI`)
- void `estimate_max_stack` (const `ModelDataStackT`< `Model` > &`im_stack`, `ParamVecT`< `Model` > &`theta_est`, `VecT` &`rllh_stack`, `CubeT` &`obsI_stack`)
- void `estimate_profile_stack` (const `ModelDataT`< `Model` > &`data`, const `IdxVecT` &`fixed_parameters`, const `MatT` &`values`, const `ParamVecT`< `Model` > &`theta_init`, `VecT` &`profile_likelihood`, `ParamVecT`< `Model` > &`profile_parameters`)
- void `clear_stats` ()
- `Model` & `get_model` ()
- void `set_model` (`Model` &`new_model`)
- `StencilT`< `Model` > `estimate_max` (const `ModelDataT`< `Model` > &`im`)
- `StencilT`< `Model` > `estimate_max` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`)
- `StencilT`< `Model` > `estimate_max` (const `ModelDataT`< `Model` > &`im`, double &`rllh`)
- `StencilT`< `Model` > `estimate_max` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, double &`rllh`)
- void `estimate_max` (const `ModelDataT`< `Model` > &`im`, `ParamT`< `Model` > &`theta`, double &`rllh`, `MatT` &`obsI`)
- void `estimate_max` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, `ParamT`< `Model` > &`theta`, double &`rllh`, `MatT` &`obsI`)
- void `estimate_max_debug` (const `ModelDataT`< `Model` > &`im`, const `ParamT`< `Model` > &`theta_init`, `ParamT`< `Model` > &`theta_est`, double &`rllh`, `MatT` &`obsI`, `MatT` &`sequence`, `VecT` &`sequence_rllh`)

Public Attributes

- double `T_init` =100.
- double `cooling_rate` =1.02
- int `num_iterations` =500

Static Public Attributes

- static constexpr int `NumExitCodes` = 9

Protected Member Functions

- `StencilT< Model > compute_estimate` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, double `&rllh`)
- `StencilT< Model > compute_estimate_debug` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, `ParamVecT< Model > &sequence`, `VecT &sequence_rllh`)
- `StencilT< Model > anneal` (const `ModelDataT< Model > &im`, const `StencilT< Model > &theta_init`, double `&rllh`, `MatT &sequence`, `VecT &sequence_rllh`)
- void `record_exit_code` (`ExitCode` code)
- virtual void `compute_estimate` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, `ParamT< Model > &theta_est`, double `&rllh`, `MatT &obsI`)
Default base class implementation computes rllh and obsI seperately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.
- virtual void `compute_profile_estimate` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, const `IdxVecT &fixed_parameters`, `ParamT< Model > &theta_est`, double `&rllh`)
- void `record_walltime` (`ClockT::time_point` start_walltime, int nimages)

Protected Attributes

- int `max_threads`
- int `num_threads`
- std::mutex `mtx`
- Model & `model`
- int `num_estimations` = 0
- double `total_walltime` = 0.
- `IdxVecT` `exit_counts`

8.46.1 Detailed Description

```
template<class Model>
class mappel::SimulatedAnnealingMaximizer< Model >
```

Definition at line 196 of file estimator.h.

8.46.2 Member Enumeration Documentation

8.46.2.1 `template<class Model> enum mappel::Estimator::ExitCode : IdxT [strong], [inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.46.3 Constructor & Destructor Documentation

8.46.3.1 `template<class Model> mappel::SimulatedAnnealingMaximizer< Model >::SimulatedAnnealingMaximizer (Model & model) [inline]`

Definition at line 202 of file estimator.h.

References `mappel::Estimator< Model >::get_debug_stats()`, and `mappel::Estimator< Model >::get_stats()`.

8.46.4 Member Function Documentation

8.46.4.1 `template<class Model> StencilT< Model > mappel::SimulatedAnnealingMaximizer< Model >::anneal (const ModelDataT< Model > & im, const StencilT< Model > & theta_init, double & rllh, MatT & sequence, VecT & sequence_rllh) [protected]`

Definition at line 1552 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::Estimator< Model >::model`, and `mappel::methods::objective::rllh()`.

8.46.4.2 `template<class Model> void mappel::ThreadedEstimator< Model >::clear_stats () [virtual], [inherited]`

Reimplemented from [mappel::Estimator< Model >](#).

Reimplemented in [mappel::IterativeMaximizer< Model >](#).

Definition at line 378 of file estimator_impl.h.

References `mappel::Estimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::mtx`, and `mappel::ThreadedEstimator< Model >::num_threads`.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`.

8.46.4.3 `template<class Model> void mappel::Estimator< Model>::compute_estimate (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta_est, double & rllh, MatT & obsl) [protected], [virtual], [inherited]`

Default base class implementation computes rllh and obsl separately from stencil. This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file estimator_impl.h.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::model`, and `mappel::methods::observed_information()`.

8.46.4.4 `template<class Model> StencilT< Model> mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, double & rllh) [protected], [virtual]`

Implements [mappel::Estimator< Model>](#).

Definition at line 1532 of file estimator_impl.h.

References `mappel::Estimator< Model>::model`, and `mappel::IterativeMaximizer< Model>::MaximizerData::rllh`.

8.46.4.5 `template<class Model> StencilT< Model> mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate_debug (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamVecT< Model> & sequence, VecT & sequence_rllh) [protected], [virtual]`

Estimators that produce a sequence of results (e.g. `IterativeEstimators`) can override this dummy debug implementation.

Reimplemented from [mappel::Estimator< Model>](#).

Definition at line 1541 of file estimator_impl.h.

References `mappel::Estimator< Model>::model`, and `mappel::IterativeMaximizer< Model>::MaximizerData::rllh`.

8.46.4.6 `template<class Model> void mappel::Estimator< Model>::compute_profile_estimate (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, const IdxVecT & fixed_parameters, ParamT< Model> & theta_est, double & rllh) [protected], [virtual], [inherited]`

Reimplemented in [mappel::IterativeMaximizer< Model>](#).

Definition at line 192 of file estimator_impl.h.

References `mappel::Estimator< Model>::model`.

Referenced by `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, and `mappel::Estimator< Model>::~~Estimator()`.

8.46.4.7 `template<class Model> StencilT< Model> mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im) [inherited]`

Definition at line 64 of file estimator_impl.h.

References `mappel::Estimator< Model>::model`.

Referenced by `mappel::Estimator< Model>::estimate_max()`, and `mappel::Estimator< Model>::~~Estimator()`.

8.46.4.8 `template<class Model> StencilT< Model> mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, const ParamT< Model> & theta_init) [inherited]`

Definition at line 82 of file estimator_impl.h.

References mappel::Estimator< Model>::estimate_max(), and mappel::methods::objective::rllh().

8.46.4.9 `template<class Model> StencilT< Model> mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, double & rllh) [inherited]`

Definition at line 73 of file estimator_impl.h.

References mappel::Estimator< Model>::estimate_max(), and mappel::Estimator< Model>::model.

8.46.4.10 `template<class Model> StencilT< Model> mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, double & rllh) [inherited]`

Definition at line 90 of file estimator_impl.h.

References mappel::Estimator< Model>::compute_estimate(), mappel::Estimator< Model>::Error, mappel::Estimator< Model>::model, mappel::print_image(), mappel::Estimator< Model>::record_exit_code(), and mappel::Estimator< Model>::record_walltime().

8.46.4.11 `template<class Model> void mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, ParamT< Model> & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 119 of file estimator_impl.h.

References mappel::Estimator< Model>::estimate_max(), and mappel::Estimator< Model>::model.

8.46.4.12 `template<class Model> void mappel::Estimator< Model>::estimate_max (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References mappel::Estimator< Model>::compute_estimate(), mappel::Estimator< Model>::Error, mappel::print_image(), mappel::Estimator< Model>::record_exit_code(), mappel::Estimator< Model>::record_walltime(), and mappel::methods::objective::rllh().

8.46.4.13 `template<class Model> void mappel::Estimator< Model>::estimate_max_debug (const ModelDataT< Model> & im, const ParamT< Model> & theta_init, ParamT< Model> & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References mappel::Estimator< Model>::compute_estimate_debug(), mappel::Estimator< Model>::model, mappel::methods::observed_information(), mappel::Estimator< Model>::record_walltime(), and mappel::methods::objective::rllh().

Referenced by mappel::Estimator< Model>::~~Estimator().

8.46.4.14 `template<class Model > void mappel::Estimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im_stack, ParamVecT< Model > & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack)`
`[inherited]`

Definition at line 170 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::estimate_max_stack()`, and `mappel::Estimator< Model >::model`.

8.46.4.15 `template<class Model > void mappel::ThreadedEstimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im, const ParamVecT< Model > & theta_init, ParamVecT< Model > & theta, VecT & rllh, CubeT & obsl)` `[virtual]`, `[inherited]`

Implements `mappel::Estimator< Model >`.

Definition at line 285 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.46.4.16 `template<class Model > void mappel::ThreadedEstimator< Model >::estimate_profile_stack (const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model > & theta_init, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters)` `[virtual]`, `[inherited]`

Implements `mappel::Estimator< Model >`.

Definition at line 323 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.46.4.17 `template<class Model > StatsT mappel::SimulatedAnnealingMaximizer< Model >::get_debug_stats ()`
`[virtual]`

Reimplemented from `mappel::ThreadedEstimator< Model >`.

Definition at line 1616 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::get_stats()`.

8.46.4.18 `template<class Model > Model & mappel::Estimator< Model >::get_model ()` `[inherited]`

Definition at line 50 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::model`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.46.4.19 `template<class Model> StatsT mappel::SimulatedAnnealingMaximizer< Model >::get_stats ()`
`[virtual]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 1604 of file `estimator_impl.h`.

References [mappel::ThreadedEstimator< Model >::get_stats\(\)](#).

8.46.4.20 `template<class Model> std::string mappel::SimulatedAnnealingMaximizer< Model >::name () const`
`[inline], [virtual]`

Implements [mappel::Estimator< Model >](#).

Definition at line 206 of file `estimator.h`.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::compute_estimate_← debug\(\)](#), and [mappel::methods::objective::rllh\(\)](#).

8.46.4.21 `template<class Model> void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code)`
`[protected], [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 386 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::model](#), [mappel::Threaded← Estimator< Model >::mtx](#), and [mappel::methods::objective::rllh\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::CGaussMLE< Model >::compute_← _estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::convergence_test\(\)](#), [mappel::Threaded← Estimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

8.46.4.22 `template<class Model> void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages)`
`[protected], [inherited]`

Definition at line 266 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::num_estimations](#), and [mappel::Estimator< Model >::total_walltime](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), [mappel::Estimator< Model >::estimate_max_← debug\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), and [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#).

8.46.4.23 `template<class Model > void mappel::Estimator< Model >::set_model (Model & new_model)` [inherited]

Definition at line 54 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::~~Estimator().

8.46.5 Member Data Documentation

8.46.5.1 `template<class Model > double mappel::SimulatedAnnealingMaximizer< Model >::cooling_rate = 1.02`

Definition at line 199 of file estimator.h.

8.46.5.2 `template<class Model > IdxVecT mappel::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 114 of file estimator.h.

Referenced by mappel::Estimator< Model >::clear_stats(), mappel::Estimator< Model >::get_stats(), and mappel::ThreadedEstimator< Model >::record_exit_code().

8.46.5.3 `template<class Model > int mappel::ThreadedEstimator< Model >::max_threads` [protected], [inherited]

Definition at line 142 of file estimator.h.

8.46.5.4 `template<class Model > Model& mappel::Estimator< Model >::model` [protected], [inherited]

Definition at line 109 of file estimator.h.

Referenced by mappel::SimulatedAnnealingMaximizer< Model >::anneal(), mappel::IterativeMaximizer< Model >::backtrack(), mappel::Estimator< Model >::compute_estimate(), mappel::CGaussMLE< Model >::compute_estimate(), mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate(), mappel::IterativeMaximizer< Model >::compute_estimate(), mappel::Estimator< Model >::compute_estimate_debug(), mappel::CGaussMLE< Model >::compute_estimate_debug(), mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug(), mappel::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::Estimator< Model >::compute_profile_estimate(), mappel::IterativeMaximizer< Model >::compute_profile_estimate(), mappel::Estimator< Model >::estimate_max(), mappel::Estimator< Model >::estimate_max_debug(), mappel::Estimator< Model >::estimate_max_stack(), mappel::ThreadedEstimator< Model >::estimate_max_stack(), mappel::ThreadedEstimator< Model >::estimate_profile_stack(), mappel::HeuristicEstimator< Model >::get_debug_stats(), mappel::Estimator< Model >::get_model(), mappel::IterativeMaximizer< Model >::local_maximize(), mappel::NewtonDiagonalMaximizer< Model >::maximize(), mappel::NewtonMaximizer< Model >::maximize(), mappel::QuasiNewtonMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >::maximize(), mappel::ThreadedEstimator< Model >::record_exit_code(), and mappel::Estimator< Model >::set_model().

8.46.5.5 `template<class Model> std::mutex mappel::ThreadedEstimator< Model >::mtx` [protected],
[inherited]

Definition at line 144 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.46.5.6 `template<class Model> int mappel::Estimator< Model >::num_estimations = 0` [protected],
[inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

8.46.5.7 `template<class Model> int mappel::SimulatedAnnealingMaximizer< Model >::num_iterations = 500`

Definition at line 200 of file estimator.h.

8.46.5.8 `template<class Model> int mappel::ThreadedEstimator< Model >::num_threads` [protected],
[inherited]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.46.5.9 `template<class Model> constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static],
[inherited]

Definition at line 39 of file estimator.h.

8.46.5.10 `template<class Model> double mappel::SimulatedAnnealingMaximizer< Model >::T_init = 100.`

Definition at line 198 of file estimator.h.

8.46.5.11 `template<class Model> double mappel::Estimator< Model >::total_walltime = 0.` [protected],
[inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

8.47 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)

8.47.1 Detailed Description

[Stencil](#) for 1D fixed-sigma models.

Definition at line 29 of file [Gauss1DModel.h](#).

8.47.2 Member Typedef Documentation

8.47.2.1 using mappel::Gauss1DModel::Stencil::ParamT = Gauss1DModel::ParamT

Definition at line 32 of file [Gauss1DModel.h](#).

8.47.3 Constructor & Destructor Documentation

8.47.3.1 mappel::Gauss1DModel::Stencil () [inline]

Definition at line 41 of file Gauss1DModel.h.

References `compute_derivatives()`.

Referenced by `mappel::Gauss1DModel::make_stencil()`.

8.47.3.2 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()`.

8.47.4 Member Function Documentation

8.47.4.1 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()`.

8.47.4.2 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()`.

8.47.4.3 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()`.

8.47.4.4 `double mappel::Gauss1DModel::Stencil::x () const` `[inline]`

Definition at line 44 of file Gauss1DModel.h.

References theta.

Referenced by Stencil().

8.47.5 Friends And Related Function Documentation

8.47.5.1 `std::ostream& operator<< (std::ostream & out, const Gauss1DModel::Stencil & s)` `[friend]`

Definition at line 164 of file Gauss1DModel.cpp.

Referenced by bg().

8.47.6 Member Data Documentation

8.47.6.1 `bool mappel::Gauss1DModel::Stencil::derivatives_computed = false`

Definition at line 31 of file Gauss1DModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

8.47.6.2 `VecT mappel::Gauss1DModel::Stencil::dx`

Definition at line 36 of file Gauss1DModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

8.47.6.3 `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().

8.47.6.4 `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().

8.47.6.5 VecT mappel::Gauss1DModel::Stencil::Gx

Definition at line 37 of file Gauss1DModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

8.47.6.6 Gauss1DModel const* mappel::Gauss1DModel::Stencil::model

Definition at line 33 of file Gauss1DModel.h.

Referenced by compute_derivatives(), and Stencil().

8.47.6.7 ParamT mappel::Gauss1DModel::Stencil::theta

Definition at line 35 of file Gauss1DModel.h.

Referenced by bg(), l(), mappel::Gauss1DModel::initial_theta_estimate(), mappel::operator<<(), and x().

8.47.6.8 VecT mappel::Gauss1DModel::Stencil::X

Definition at line 38 of file Gauss1DModel.h.

Referenced by mappel::operator<<(), mappel::Gauss1DModel::pixel_grad(), mappel::Gauss1DModel::pixel_model_value(), and Stencil().

The documentation for this class was generated from the following files:

- [Gauss1DModel.h](#)
- [Gauss1DModel.cpp](#)

8.48 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)

8.48.1 Detailed Description

`Stencil` for 1D variable-sigma models.

Definition at line 24 of file `Gauss1DsModel.h`.

8.48.2 Member Typedef Documentation

8.48.2.1 using `mappel::Gauss1DsModel::Stencil::ParamT` = `Gauss1DsModel::ParamT`

Definition at line 27 of file `Gauss1DsModel.h`.

8.48.3 Constructor & Destructor Documentation

8.48.3.1 `mappel::Gauss1DsModel::Stencil::Stencil ()` [inline]

Definition at line 38 of file `Gauss1DsModel.h`.

References `compute_derivatives()`.

Referenced by `mappel::Gauss1DsModel::make_stencil()`.

8.48.3.2 `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()`.

8.48.4 Member Function Documentation

8.48.4.1 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().

8.48.4.2 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().

8.48.4.3 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().

8.48.4.4 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().

8.48.4.5 double mappel::Gauss1DsModel::Stencil::x () const [inline]

Definition at line 41 of file Gauss1DsModel.h.

References theta.

Referenced by Stencil().

8.48.5 Friends And Related Function Documentation

8.48.5.1 `std::ostream& operator<< (std::ostream & out, const Gauss1DsModel::Stencil & s)` [friend]

Definition at line 182 of file Gauss1DsModel.cpp.

Referenced by `sigma()`.

8.48.6 Member Data Documentation

8.48.6.1 `bool mappel::Gauss1DsModel::Stencil::derivatives_computed = false`

Definition at line 26 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, and `mappel::operator<<()`.

8.48.6.2 `VecT mappel::Gauss1DsModel::Stencil::dx`

Definition at line 31 of file Gauss1DsModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, and `Stencil()`.

8.48.6.3 `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()`.

8.48.6.4 `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()`.

8.48.6.5 `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()`.

8.48.6.6 `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()`.

8.48.6.7 VecT mappel::Gauss1DsModel::Stencil::Gx

Definition at line 32 of file Gauss1DsModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

8.48.6.8 Gauss1DsModel const* mappel::Gauss1DsModel::Stencil::model

Definition at line 28 of file Gauss1DsModel.h.

Referenced by compute_derivatives(), and Stencil().

8.48.6.9 ParamT mappel::Gauss1DsModel::Stencil::theta

Definition at line 30 of file Gauss1DsModel.h.

Referenced by bg(), l(), mappel::operator<<(), sigma(), and x().

8.48.6.10 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](#)

8.49 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)

8.49.1 Detailed Description

[Stencil](#) for 2D free-sigma (astigmatic) models.

Definition at line 32 of file [Gauss2DsxxyModel.h](#).

8.49.2 Member Typedef Documentation**8.49.2.1 typedef [Gauss2DsxxyModel::ParamT](#) [mappel::Gauss2DsxxyModel::Stencil::ParamT](#)**

Definition at line 35 of file [Gauss2DsxxyModel.h](#).

8.49.3 Constructor & Destructor Documentation

8.49.3.1 mappel::Gauss2DsxxyModel::Stencil::Stencil () [inline]

Definition at line 47 of file Gauss2DsxxyModel.h.

References `compute_derivatives()`.

Referenced by `mappel::Gauss2DsxxyModel::make_stencil()`.

8.49.3.2 mappel::Gauss2DsxxyModel::Stencil::Stencil (const Gauss2DsxxyModel & model, const ParamT & theta, bool _compute_derivatives = true)

8.49.4 Member Function Documentation

8.49.4.1 double mappel::Gauss2DsxxyModel::Stencil::bg () const [inline]

Definition at line 53 of file Gauss2DsxxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxxyModel::pixel_model_value()`.

8.49.4.2 void mappel::Gauss2DsxxyModel::Stencil::compute_derivatives ()

Referenced by `Stencil()`.

8.49.4.3 double mappel::Gauss2DsxxyModel::Stencil::l () const [inline]

Definition at line 52 of file Gauss2DsxxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad()`, `mappel::Gauss2DsxxyModel::pixel_grad2()`, `mappel::Gauss2DsxxyModel::pixel_hess()`, and `mappel::Gauss2DsxxyModel::pixel_model_value()`.

8.49.4.4 double mappel::Gauss2DsxxyModel::Stencil::sigmaX () const [inline]

Definition at line 54 of file Gauss2DsxxyModel.h.

References `theta`.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad2()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

8.49.4.5 double mappel::Gauss2DsxxyModel::Stencil::sigmaY () const [inline]

Definition at line 55 of file Gauss2DsxxyModel.h.

References `operator<<`, and `theta`.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad2()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

8.49.4.6 `double mappel::Gauss2DsxyModel::Stencil::x () const [inline]`

Definition at line 50 of file Gauss2DsxyModel.h.

References `theta`.

8.49.4.7 `double mappel::Gauss2DsxyModel::Stencil::y () const [inline]`

Definition at line 51 of file Gauss2DsxyModel.h.

References `theta`.

8.49.5 Friends And Related Function Documentation

8.49.5.1 `std::ostream& operator<< (std::ostream & out, const Gauss2DsxyModel::Stencil & s) [friend]`

Referenced by `sigmaY()`.

8.49.6 Member Data Documentation

8.49.6.1 `bool mappel::Gauss2DsxyModel::Stencil::derivatives_computed =false`

Definition at line 34 of file Gauss2DsxyModel.h.

8.49.6.2 `VecT mappel::Gauss2DsxyModel::Stencil::dx`

Definition at line 39 of file Gauss2DsxyModel.h.

8.49.6.3 `VecT mappel::Gauss2DsxyModel::Stencil::DX`

Definition at line 42 of file Gauss2DsxyModel.h.

Referenced by `mappel::Gauss2DsxyModel::pixel_grad()`, and `mappel::Gauss2DsxyModel::pixel_hess()`.

8.49.6.4 `VecT mappel::Gauss2DsxyModel::Stencil::DXS`

Definition at line 44 of file Gauss2DsxyModel.h.

Referenced by `mappel::Gauss2DsxyModel::pixel_grad()`, `mappel::Gauss2DsxyModel::pixel_grad2()`, and `mappel::Gauss2DsxyModel::pixel_hess()`.

8.49.6.5 `VecT mappel::Gauss2DsxyModel::Stencil::DXS2`

Definition at line 45 of file Gauss2DsxyModel.h.

Referenced by `mappel::Gauss2DsxyModel::pixel_grad2()`, and `mappel::Gauss2DsxyModel::pixel_hess()`.

8.49.6.6 VecT mappel::Gauss2DsxxyModel::Stencil::DXSX

Definition at line 43 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel_hess().

8.49.6.7 VecT mappel::Gauss2DsxxyModel::Stencil::dy

Definition at line 39 of file Gauss2DsxxyModel.h.

8.49.6.8 VecT mappel::Gauss2DsxxyModel::Stencil::DY

Definition at line 42 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel_grad(), and mappel::Gauss2DsxxyModel::pixel_hess().

8.49.6.9 VecT mappel::Gauss2DsxxyModel::Stencil::DYS

Definition at line 44 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel_grad(), mappel::Gauss2DsxxyModel::pixel_grad2(), and mappel::Gauss2DsxxyModel::pixel_hess().

8.49.6.10 VecT mappel::Gauss2DsxxyModel::Stencil::DYS2

Definition at line 45 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel_grad2(), and mappel::Gauss2DsxxyModel::pixel_hess().

8.49.6.11 VecT mappel::Gauss2DsxxyModel::Stencil::DYSX

Definition at line 43 of file Gauss2DsxxyModel.h.

8.49.6.12 VecT mappel::Gauss2DsxxyModel::Stencil::DYSY

Definition at line 46 of file Gauss2DsxxyModel.h.

Referenced by mappel::Gauss2DsxxyModel::pixel_hess().

8.49.6.13 VecT mappel::Gauss2DsxxyModel::Stencil::Gx

Definition at line 40 of file Gauss2DsxxyModel.h.

8.49.6.14 VecT mappel::Gauss2DsxxyModel::Stencil::Gy

Definition at line 40 of file Gauss2DsxxyModel.h.

8.49.6.15 `Gauss2DsxModel` `const*` `mappel::Gauss2DsxModel::Stencil::model`

Definition at line 36 of file `Gauss2DsxModel.h`.

8.49.6.16 `ParamT` `mappel::Gauss2DsxModel::Stencil::theta`

Definition at line 38 of file `Gauss2DsxModel.h`.

Referenced by `bg()`, `l()`, `sigmaX()`, `sigmaY()`, `x()`, and `y()`.

8.49.6.17 `VecT` `mappel::Gauss2DsxModel::Stencil::X`

Definition at line 41 of file `Gauss2DsxModel.h`.

Referenced by `mappel::Gauss2DsxModel::pixel_grad()`, `mappel::Gauss2DsxModel::pixel_grad2()`, `mappel::Gauss2DsxModel::pixel_hess()`, and `mappel::Gauss2DsxModel::pixel_model_value()`.

8.49.6.18 `VecT` `mappel::Gauss2DsxModel::Stencil::Y`

Definition at line 41 of file `Gauss2DsxModel.h`.

Referenced by `mappel::Gauss2DsxModel::pixel_grad()`, `mappel::Gauss2DsxModel::pixel_grad2()`, `mappel::Gauss2DsxModel::pixel_hess()`, and `mappel::Gauss2DsxModel::pixel_model_value()`.

The documentation for this class was generated from the following file:

- [Gauss2DsxModel.h](#)

8.50 `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)

8.50.1 Detailed Description

[Stencil](#) for 2D scalar-sigma models.

Definition at line 29 of file [Gauss2DsModel.h](#).

8.50.2 Member Typedef Documentation

8.50.2.1 typedef [Gauss2DsModel::ParamT](#) mappel::Gauss2DsModel::Stencil::ParamT

Definition at line 32 of file [Gauss2DsModel.h](#).

8.50.3 Constructor & Destructor Documentation

8.50.3.1 mappel::Gauss2DsModel::Stencil::Stencil () [inline]

Definition at line 43 of file [Gauss2DsModel.h](#).

References [compute_derivatives\(\)](#).

Referenced by [mappel::Gauss2DsModel::make_stencil\(\)](#).

8.50.3.2 `mappel::Gauss2DsModel::Stencil::Stencil (const Gauss2DsModel & model, const ParamT & theta, bool
_compute_derivatives = true)`

Definition at line 286 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()`.

8.50.4 Member Function Documentation

8.50.4.1 `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()`.

8.50.4.2 `void mappel::Gauss2DsModel::Stencil::compute_derivatives ()`

Definition at line 300 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()`.

8.50.4.3 `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()`.

8.50.4.4 `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()`.

8.50.4.5 `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().

8.50.4.6 `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().

8.50.4.7 `double mappel::Gauss2DsModel::Stencil::x () const [inline]`

Definition at line 46 of file Gauss2DsModel.h.

References theta.

Referenced by Stencil().

8.50.4.8 `double mappel::Gauss2DsModel::Stencil::y () const [inline]`

Definition at line 47 of file Gauss2DsModel.h.

References theta.

Referenced by Stencil().

8.50.5 Friends And Related Function Documentation

8.50.5.1 `std::ostream& operator<< (std::ostream & out, const Gauss2DsModel::Stencil & s) [friend]`

Definition at line 318 of file Gauss2DsModel.cpp.

Referenced by sigmaY().

8.50.6 Member Data Documentation

8.50.6.1 `bool mappel::Gauss2DsModel::Stencil::derivatives_computed =false`

Definition at line 31 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

8.50.6.2 VecT mappel::Gauss2DsModel::Stencil::dx

Definition at line 36 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

8.50.6.3 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().

8.50.6.4 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().

8.50.6.5 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().

8.50.6.6 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().

8.50.6.7 VecT mappel::Gauss2DsModel::Stencil::dy

Definition at line 36 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

8.50.6.8 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().

8.50.6.9 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().

8.50.6.10 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().

8.50.6.11 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().

8.50.6.12 VecT mappel::Gauss2DsModel::Stencil::Gx

Definition at line 37 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

8.50.6.13 VecT mappel::Gauss2DsModel::Stencil::Gy

Definition at line 37 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

8.50.6.14 Gauss2DsModel const* mappel::Gauss2DsModel::Stencil::model

Definition at line 33 of file Gauss2DsModel.h.

Referenced by compute_derivatives(), and Stencil().

8.50.6.15 ParamT mappel::Gauss2DsModel::Stencil::theta

Definition at line 35 of file Gauss2DsModel.h.

Referenced by bg(), l(), mappel::operator<<(), sigma_ratio(), x(), and y().

8.50.6.16 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().

8.50.6.17 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](#)

8.51 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)

8.51.1 Detailed Description

[Stencil](#) for 2D fixed-sigma models.

Definition at line 26 of file [Gauss2DModel.h](#).

8.51.2 Member Typedef Documentation

8.51.2.1 using mappel::Gauss2DModel::Stencil::ParamT = Gauss2DModel::ParamT

Definition at line 29 of file [Gauss2DModel.h](#).

8.51.3 Constructor & Destructor Documentation

8.51.3.1 mappel::Gauss2DModel::Stencil::Stencil() [inline]

Definition at line 38 of file [Gauss2DModel.h](#).

References [compute_derivatives\(\)](#).

Referenced by [mappel::Gauss2DModel::make_stencil\(\)](#).

8.51.3.2 `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()`.

8.51.4 Member Function Documentation

8.51.4.1 `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()`.

8.51.4.2 `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()`.

8.51.4.3 `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()`.

8.51.4.4 `double mappel::Gauss2DModel::Stencil::x () const [inline]`

Definition at line 41 of file Gauss2DModel.h.

References `theta`.

Referenced by `Stencil()`.

8.51.4.5 `double mappel::Gauss2DModel::Stencil::y () const` `[inline]`

Definition at line 42 of file Gauss2DModel.h.

References `theta`.

Referenced by `Stencil()`.

8.51.5 Friends And Related Function Documentation

8.51.5.1 `std::ostream& operator<< (std::ostream & out, const Gauss2DModel::Stencil & s)` `[friend]`

Definition at line 249 of file Gauss2DModel.cpp.

Referenced by `bg()`.

8.51.6 Member Data Documentation

8.51.6.1 `bool mappel::Gauss2DModel::Stencil::derivatives_computed =false`

Definition at line 28 of file Gauss2DModel.h.

Referenced by `compute_derivatives()`, and `mappel::operator<<()`.

8.51.6.2 `VecT mappel::Gauss2DModel::Stencil::dx`

Definition at line 33 of file Gauss2DModel.h.

Referenced by `compute_derivatives()`, `mappel::operator<<()`, and `Stencil()`.

8.51.6.3 `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()`.

8.51.6.4 `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()`.

8.51.6.5 VecT mappel::Gauss2DModel::Stencil::dy

Definition at line 33 of file Gauss2DModel.h.

Referenced by compute_derivatives(), mappel::operator<<(), and Stencil().

8.51.6.6 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().

8.51.6.7 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().

8.51.6.8 VecT mappel::Gauss2DModel::Stencil::Gx

Definition at line 34 of file Gauss2DModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

8.51.6.9 VecT mappel::Gauss2DModel::Stencil::Gy

Definition at line 34 of file Gauss2DModel.h.

Referenced by compute_derivatives(), and mappel::operator<<().

8.51.6.10 Gauss2DModel const* mappel::Gauss2DModel::Stencil::model

Definition at line 30 of file Gauss2DModel.h.

Referenced by compute_derivatives(), and Stencil().

8.51.6.11 ParamT mappel::Gauss2DModel::Stencil::theta

Definition at line 32 of file Gauss2DModel.h.

Referenced by bg(), l(), mappel::operator<<(), x(), and y().

8.51.6.12 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().

8.51.6.13 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().

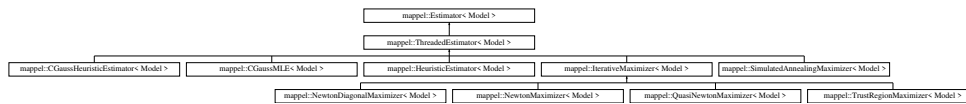
The documentation for this class was generated from the following files:

- [Gauss2DModel.h](#)
- [Gauss2DModel.cpp](#)

8.52 mappel::ThreadedEstimator< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for mappel::ThreadedEstimator< Model >:



Public Types

- enum [ExitCode](#) : IdxT {
[ExitCode::Unassigned](#) = 8, [ExitCode::MaxIter](#) = 7, [ExitCode::MaxBacktracks](#) = 6, [ExitCode::TrustRegionRadius](#) = 5,
[ExitCode::GradRatio](#) = 4, [ExitCode::FunctionChange](#) = 3, [ExitCode::StepSize](#) = 2, [ExitCode::Success](#) = 1,
[ExitCode::Error](#) = 0 }

Public Member Functions

- [ThreadedEstimator](#) (Model &model)
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im, const [ParamVecT](#)< Model > &theta_init, [ParamVecT](#)< Model > &theta, [VecT](#) &rllh, [CubeT](#) &obs1)
- void [estimate_profile_stack](#) (const [ModelDataT](#)< Model > &data, const [IdxVecT](#) &fixed_parameters, const [MatT](#) &values, const [ParamVecT](#)< Model > &theta_init, [VecT](#) &profile_likelihood, [ParamVecT](#)< Model > &profile←parameters)
- [StatsT](#) [get_stats](#) ()
- [StatsT](#) [get_debug_stats](#) ()
- void [clear_stats](#) ()
- virtual std::string [name](#) () const =0
- Model & [get_model](#) ()
- void [set_model](#) (Model &new_model)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im)
- [StencilT](#)< Model > [estimate_max](#) (const [ModelDataT](#)< Model > &im, const [ParamT](#)< Model > &theta_init)

- [StencilT](#)< [Model](#) > [estimate_max](#) (const [ModelDataT](#)< [Model](#) > &im, double &rllh)
- [StencilT](#)< [Model](#) > [estimate_max](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, double &rllh)
- void [estimate_max](#) (const [ModelDataT](#)< [Model](#) > &im, [ParamT](#)< [Model](#) > &theta, double &rllh, [MatT](#) &obsI)
- void [estimate_max](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, [ParamT](#)< [Model](#) > &theta, double &rllh, [MatT](#) &obsI)
- void [estimate_max_debug](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, [ParamT](#)< [Model](#) > &theta_est, double &rllh, [MatT](#) &obsI, [MatT](#) &sequence, [VecT](#) &sequence_rllh)
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< [Model](#) > &im_stack, [ParamVecT](#)< [Model](#) > &theta_est_← stack, [VecT](#) &rllh_stack, [CubeT](#) &obsI_stack)

Static Public Attributes

- static constexpr int [NumExitCodes](#) = 9

Protected Member Functions

- void [record_exit_code](#) ([ExitCode](#) code)
 - virtual [StencilT](#)< [Model](#) > [compute_estimate](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, double &rllh)=0
 - virtual void [compute_estimate](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, [ParamT](#)< [Model](#) > &theta_est, double &rllh, [MatT](#) &obsI)
- Default base class implementation computes rllh and obsI separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.*
- virtual [StencilT](#)< [Model](#) > [compute_estimate_debug](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, [ParamVecT](#)< [Model](#) > &sequence, [VecT](#) &sequence_rllh)
 - virtual void [compute_profile_estimate](#) (const [ModelDataT](#)< [Model](#) > &im, const [ParamT](#)< [Model](#) > &theta_init, const [IdxVecT](#) &fixed_parameters, [ParamT](#)< [Model](#) > &theta_est, double &rllh)
 - void [record_walltime](#) ([ClockT](#)::time_point start_walltime, int nimages)

Protected Attributes

- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- [Model](#) & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.
- [IdxVecT](#) [exit_counts](#)

8.52.1 Detailed Description

```
template<class Model>
class mappel::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 127 of file `estimator.h`.

8.52.2 Member Enumeration Documentation

8.52.2.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT [strong], [inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.52.3 Constructor & Destructor Documentation

8.52.3.1 `template<class Model > mappel::ThreadedEstimator< Model >::ThreadedEstimator (Model & model)`

Definition at line 276 of file estimator_impl.h.

8.52.4 Member Function Documentation

8.52.4.1 `template<class Model > void mappel::ThreadedEstimator< Model >::clear_stats () [virtual]`Reimplemented from [mappel::Estimator< Model >](#).Reimplemented in [mappel::IterativeMaximizer< Model >](#).

Definition at line 378 of file estimator_impl.h.

References [mappel::Estimator< Model >::clear_stats\(\)](#), [mappel::ThreadedEstimator< Model >::mtx](#), and [mappel::ThreadedEstimator< Model >::num_threads](#).Referenced by [mappel::IterativeMaximizer< Model >::clear_stats\(\)](#).8.52.4.2 `template<class Model > virtual StencilT<Model> mappel::Estimator< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [protected], [pure virtual], [inherited]`Implemented in [mappel::IterativeMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), and [mappel::CGaussMLE< Model >](#).Referenced by [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::compute_estimate←_debug\(\)](#), [mappel::Estimator< Model >::estimate_max\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max←_stack\(\)](#), [mappel::HeuristicEstimator< Model >::name\(\)](#), [mappel::CGaussHeuristicEstimator< Model >::name\(\)](#), [mappel::CGaussMLE< Model >::name\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::name\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.52.4.3 `template<class Model> void mappel::Estimator< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl) [protected], [virtual], [inherited]`

Default base class implementation computes rllh and obsl separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::model](#), and [mappel::methods::observed_information\(\)](#).

8.52.4.4 `template<class Model> StencilT< Model > mappel::Estimator< Model >::compute_estimate_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamVecT< Model > & sequence, VecT & sequence_rllh) [protected], [virtual], [inherited]`

Estimators that produce a sequence of results (e.g. [IterativeEstimators](#)) can override this dummy debug implementation.

Reimplemented in [mappel::IterativeMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), and [mappel::CGaussMLE< Model >](#).

Definition at line 242 of file estimator_impl.h.

References [mappel::Estimator< Model >::compute_estimate\(\)](#), [mappel::Estimator< Model >::Error](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), and [mappel::Estimator< Model >::record_exit_code\(\)](#).

Referenced by [mappel::Estimator< Model >::estimate_max_debug\(\)](#), [mappel::CGaussMLE< Model >::name\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::name\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.52.4.5 `template<class Model> void mappel::Estimator< Model >::compute_profile_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT< Model > & theta_est, double & rllh) [protected], [virtual], [inherited]`

Reimplemented in [mappel::IterativeMaximizer< Model >](#).

Definition at line 192 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.52.4.6 `template<class Model> StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im) [inherited]`

Definition at line 64 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), and [mappel::Estimator< Model >::~Estimator\(\)](#).

8.52.4.7 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init) [inherited]`

Definition at line 82 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::methods::objective::rllh()`.

8.52.4.8 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh) [inherited]`

Definition at line 73 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.52.4.9 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [inherited]`

Definition at line 90 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.52.4.10 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 119 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max()`, and `mappel::Estimator< Model >::model`.

8.52.4.11 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::print_image()`, `mappel::Estimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

8.52.4.12 `template<class Model > void mappel::Estimator< Model >::estimate_max_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.52.4.13 `template<class Model> void mappel::Estimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im_stack, ParamVecT< Model> & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack)`
`[inherited]`

Definition at line 170 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::estimate_max_stack()`, and `mappel::Estimator< Model>::model`.

8.52.4.14 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_max_stack (const ModelDataStackT< Model> & im, const ParamVecT< Model> & theta_init, ParamVecT< Model> & theta, VecT & rllh, CubeT & obsl)` `[virtual]`

Implements `mappel::Estimator< Model>`.

Definition at line 285 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, `mappel::Estimator< Model>::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.52.4.15 `template<class Model> void mappel::ThreadedEstimator< Model>::estimate_profile_stack (const ModelDataT< Model> & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model> & theta_init, VecT & profile_likelihood, ParamVecT< Model> & profile_parameters)` `[virtual]`

Implements `mappel::Estimator< Model>`.

Definition at line 323 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::compute_profile_estimate()`, `mappel::Estimator< Model>::Error`, `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model>::record_exit_code()`, and `mappel::Estimator< Model>::record_walltime()`.

8.52.4.16 `template<class Model> StatsT mappel::ThreadedEstimator< Model>::get_debug_stats ()` `[virtual]`

Implements `mappel::Estimator< Model>`.

Reimplemented in `mappel::IterativeMaximizer< Model>`, `mappel::SimulatedAnnealingMaximizer< Model>`, `mappel::CGaussMLE< Model>`, `mappel::CGaussHeuristicEstimator< Model>`, and `mappel::HeuristicEstimator< Model>`.

Definition at line 372 of file `estimator_impl.h`.

References `mappel::ThreadedEstimator< Model>::get_stats()`.

8.52.4.17 `template<class Model> Model & mappel::Estimator< Model>::get_model ()` `[inherited]`

Definition at line 50 of file `estimator_impl.h`.

References `mappel::Estimator< Model>::model`.

Referenced by `mappel::Estimator< Model>::~~Estimator()`.

8.52.4.18 `template<class Model> StatsT mappel::ThreadedEstimator< Model >::get_stats () [virtual]`

Reimplemented from [mappel::Estimator< Model >](#).

Reimplemented in [mappel::IterativeMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), [mappel::CGaussMLE< Model >](#), [mappel::CGaussHeuristicEstimator< Model >](#), and [mappel::HeuristicEstimator< Model >](#).

Definition at line 361 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::get_stats\(\)](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::Estimator< Model >::num_estimations](#), [mappel::ThreadedEstimator< Model >::num_threads](#), and [mappel::Estimator< Model >::total_walltime](#).

Referenced by [mappel::ThreadedEstimator< Model >::get_debug_stats\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::CGaussHeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::CGaussMLE< Model >::get_debug_stats\(\)](#), [mappel::HeuristicEstimator< Model >::get_stats\(\)](#), [mappel::CGaussHeuristicEstimator< Model >::get_stats\(\)](#), [mappel::CGaussMLE< Model >::get_stats\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::get_stats\(\)](#), and [mappel::IterativeMaximizer< Model >::get_stats\(\)](#).

8.52.4.19 `template<class Model> virtual std::string mappel::Estimator< Model >::name () const [pure virtual], [inherited]`

Implemented in [mappel::TrustRegionMaximizer< Model >](#), [mappel::QuasiNewtonMaximizer< Model >](#), [mappel::NewtonMaximizer< Model >](#), [mappel::NewtonDiagonalMaximizer< Model >](#), [mappel::SimulatedAnnealingMaximizer< Model >](#), [mappel::CGaussMLE< Model >](#), [mappel::CGaussHeuristicEstimator< Model >](#), and [mappel::HeuristicEstimator< Model >](#).

Referenced by [mappel::Estimator< Model >::~~Estimator\(\)](#).

8.52.4.20 `template<class Model> void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code) [protected], [virtual]`

Implements [mappel::Estimator< Model >](#).

Definition at line 386 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::mtx](#), and [mappel::methods::objective::rllh\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::convergence_test\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

8.52.4.21 `template<class Model> void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages) [protected], [inherited]`

Definition at line 266 of file `estimator_impl.h`.

References [mappel::Estimator< Model >::num_estimations](#), and [mappel::Estimator< Model >::total_walltime](#).

Referenced by [mappel::Estimator< Model >::estimate_max\(\)](#), [mappel::Estimator< Model >::estimate_max_debug\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), and [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#).

8.52.4.22 `template<class Model > void mappel::Estimator< Model >::set_model (Model & new_model)` [inherited]

Definition at line 54 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::~~Estimator().

8.52.5 Member Data Documentation

8.52.5.1 `template<class Model > IdxVecT mappel::Estimator< Model >::exit_counts` [protected], [inherited]

Definition at line 114 of file estimator.h.

Referenced by mappel::Estimator< Model >::clear_stats(), mappel::Estimator< Model >::get_stats(), and mappel::ThreadedEstimator< Model >::record_exit_code().

8.52.5.2 `template<class Model > int mappel::ThreadedEstimator< Model >::max_threads` [protected]

Definition at line 142 of file estimator.h.

8.52.5.3 `template<class Model > Model& mappel::Estimator< Model >::model` [protected], [inherited]

Definition at line 109 of file estimator.h.

Referenced by mappel::SimulatedAnnealingMaximizer< Model >::anneal(), mappel::IterativeMaximizer< Model >::backtrack(), mappel::Estimator< Model >::compute_estimate(), mappel::CGaussMLE< Model >::compute_estimate(), mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate(), mappel::IterativeMaximizer< Model >::compute_estimate(), mappel::Estimator< Model >::compute_estimate_debug(), mappel::CGaussMLE< Model >::compute_estimate_debug(), mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug(), mappel::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::Estimator< Model >::compute_profile_estimate(), mappel::IterativeMaximizer< Model >::compute_profile_estimate(), mappel::Estimator< Model >::estimate_max(), mappel::Estimator< Model >::estimate_max_debug(), mappel::Estimator< Model >::estimate_max_stack(), mappel::ThreadedEstimator< Model >::estimate_max_stack(), mappel::ThreadedEstimator< Model >::estimate_profile_stack(), mappel::HeuristicEstimator< Model >::get_debug_stats(), mappel::Estimator< Model >::get_model(), mappel::IterativeMaximizer< Model >::local_maximize(), mappel::NewtonDiagonalMaximizer< Model >::maximize(), mappel::NewtonMaximizer< Model >::maximize(), mappel::QuasiNewtonMaximizer< Model >::maximize(), mappel::TrustRegionMaximizer< Model >::maximize(), mappel::ThreadedEstimator< Model >::record_exit_code(), and mappel::Estimator< Model >::set_model().

8.52.5.4 `template<class Model > std::mutex mappel::ThreadedEstimator< Model >::mtx` [protected]

Definition at line 144 of file estimator.h.

Referenced by mappel::ThreadedEstimator< Model >::clear_stats(), mappel::IterativeMaximizer< Model >::clear_stats(), mappel::ThreadedEstimator< Model >::get_stats(), mappel::IterativeMaximizer< Model >::get_stats(), mappel::ThreadedEstimator< Model >::record_exit_code(), and mappel::IterativeMaximizer< Model >::record_statistics().

8.52.5.5 `template<class Model> int mappel::Estimator< Model >::num_estimations = 0` [protected],
[inherited]

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

8.52.5.6 `template<class Model> int mappel::ThreadedEstimator< Model >::num_threads` [protected]

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.52.5.7 `template<class Model> constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static],
[inherited]

Definition at line 39 of file estimator.h.

8.52.5.8 `template<class Model> double mappel::Estimator< Model >::total_walltime = 0.` [protected],
[inherited]

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

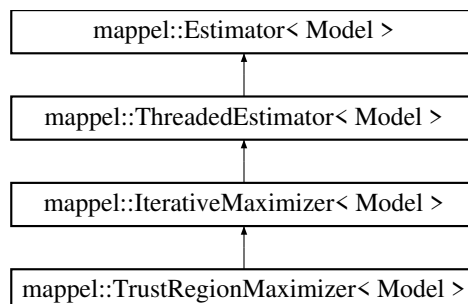
The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

8.53 mappel::TrustRegionMaximizer< Model > Class Template Reference

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/estimator.h>
```

Inheritance diagram for `mappel::TrustRegionMaximizer< Model >`:



Public Types

- using `MaximizerData` = typename `IterativeMaximizer< Model >::MaximizerData`
- enum `ExitCode` : `IdxT` {
`ExitCode::Unassigned` = 8, `ExitCode::MaxIter` = 7, `ExitCode::MaxBacktracks` = 6, `ExitCode::TrustRegionRadius` = 5,
`ExitCode::GradRatio` = 4, `ExitCode::FunctionChange` = 3, `ExitCode::StepSize` = 2, `ExitCode::Success` = 1,
`ExitCode::Error` = 0 }

Public Member Functions

- `TrustRegionMaximizer` (`Model &model`, `int max_iterations=DEFAULT_ITERATIONS`)
- `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` ()
- `void local_maximize` (const `ModelDataT< Model > &im`, const `StencilT< Model > &theta_init`, `StencilT< Model > &stencil`, `double &rllh`)
Perform a local maximization to finish off a simulated annealing run.
- `void estimate_max_stack` (const `ModelDataStackT< Model > &im`, const `ParamVecT< Model > &theta_init`, `ParamVecT< Model > &theta`, `VecT &rllh`, `CubeT &obsI`)
- `void estimate_max_stack` (const `ModelDataStackT< Model > &im_stack`, `ParamVecT< Model > &theta_est`, `VecT &rllh_stack`, `CubeT &obsI_stack`)
- `void estimate_profile_stack` (const `ModelDataT< Model > &data`, const `IdxVecT &fixed_parameters`, const `MatT &values`, const `ParamVecT< Model > &theta_init`, `VecT &profile_likelihood`, `ParamVecT< Model > &profile_parameters`)
- `Model &get_model` ()
- `void set_model` (`Model &new_model`)
- `StencilT< Model > estimate_max` (const `ModelDataT< Model > &im`)
- `StencilT< Model > estimate_max` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`)
- `StencilT< Model > estimate_max` (const `ModelDataT< Model > &im`, `double &rllh`)
- `StencilT< Model > estimate_max` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, `double &rllh`)
- `void estimate_max` (const `ModelDataT< Model > &im`, `ParamT< Model > &theta`, `double &rllh`, `MatT &obsI`)
- `void estimate_max` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, `ParamT< Model > &theta`, `double &rllh`, `MatT &obsI`)
- `void estimate_max_debug` (const `ModelDataT< Model > &im`, const `ParamT< Model > &theta_init`, `ParamT< Model > &theta_est`, `double &rllh`, `MatT &obsI`, `MatT &sequence`, `VecT &sequence_rllh`)

Static Public Attributes

- static const double `rho_cauchy_min` = 0.1
- static const double `rho_obj_min` = 0.25
- static const double `rho_obj_opt` = 0.75
- static const double `delta_decrease_min` = 0.125
- static const double `delta_decrease` = 0.25

- static const double `delta_increase` = 2
- static const double `min_scaling` = 1.0e-5
- static const double `max_scaling` = 1.0e5
- static const double `delta_init_min` = 1.0e-3
- static const double `delta_init_max` = 1.0e3
- static const double `boundary_stepback_min_kappa` = 1.0 - 1.0e-5
- static constexpr int `NumExitCodes` = 9

Protected Member Functions

- void `maximize` (`MaximizerData` &data)
 - `VecT` `bound_step` (const `VecT` &step_hat, const `VecT` &D, const `VecT` &theta, const `VecT` &lbound, const `VecT` &ubound)
 - void `record_run_statistics` (const `MaximizerData` &data)
 - `StencilT`< `Model` > `compute_estimate` (const `ModelDataT`< `Model` > &im, const `ParamT`< `Model` > &theta_init, double &rllh)
 - virtual void `compute_estimate` (const `ModelDataT`< `Model` > &im, const `ParamT`< `Model` > &theta_init, `ParamT`< `Model` > &theta_est, double &rllh, `MatT` &obsI)
- Default base class implementation computes rllh and obsI separately from stencil This should be overridden by `Estimator` subclasses that already have access to this information.*
- `StencilT`< `Model` > `compute_estimate_debug` (const `ModelDataT`< `Model` > &im, const `ParamT`< `Model` > &theta_init, `ParamVecT`< `Model` > &sequence, `VecT` &sequence_rllh)
 - void `compute_profile_estimate` (const `ModelDataT`< `Model` > &im, const `ParamT`< `Model` > &theta_init, const `IdxVecT` &fixed_parameters, `ParamT`< `Model` > &theta_est, double &rllh)
 - virtual void `maximize` (`MaximizerData` &data)=0
 - bool `backtrack` (`MaximizerData` &data)
 - bool `convergence_test` (`MaximizerData` &data)
 - void `record_exit_code` (`ExitCode` code)
 - void `record_walltime` (`ClockT`::time_point start_walltime, int nimages)

Static Protected Member Functions

- static `VecT` `compute_D_scale` (const `VecT` &oldDscale, const `VecT` &grad2)
 - static double `compute_initial_trust_radius` (const `VecT` &ghat)
 - static double `quadratic_model_value` (const `VecT` &step, const `VecT` &grad, const `MatT` &hess)
- Quadratic model value at given step Compute a quadratic model.*
- static void `compute_bound_scaling_vec` (const `VecT` &theta, const `VecT` &grad, const `VecT` &lbound, const `VecT` &ubound, `VecT` &v, `VecT` &Jv)
- The vector used for bound constrained TR scaling.*
- static `VecT` `compute_cauchy_point` (const `VecT` &g, const `MatT` &H, double delta)
 - static `VecT` `solve_TR_subproblem` (const `VecT` &g, const `MatT` &H, double delta, double epsilon)
- Exactly solver the TR subproblem even for non-positive definite H.*
- static `VecT` `solve_restricted_step_length_newton` (const `VecT` &g, const `MatT` &H, double delta, double lambda←_lb, double lambda_ub, double epsilon)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = 1e-6
- double [delta](#) = 1e-6
- double [lambda_min](#) = 0.05
- double [alpha](#) = 1e-4
- int [max_backtracks](#) = 8
- int [total_iterations](#) = 0
- int [total_backtracks](#) = 0
- int [total_fun_evals](#) = 0
- int [total_der_evals](#) = 0
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.
- [IdxVecT](#) [exit_counts](#)

8.53.1 Detailed Description

```
template<class Model>
class mappel::TrustRegionMaximizer< Model >
```

Definition at line 375 of file estimator.h.

8.53.2 Member Typedef Documentation

8.53.2.1 `template<class Model > using mappel::TrustRegionMaximizer< Model >::MaximizerData = typename IterativeMaximizer<Model>::MaximizerData`

Definition at line 377 of file estimator.h.

8.53.3 Member Enumeration Documentation

8.53.3.1 `template<class Model > enum mappel::Estimator::ExitCode : IdxT [strong], [inherited]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Success
Error

Definition at line 40 of file estimator.h.

8.53.4 Constructor & Destructor Documentation

8.53.4.1 `template<class Model> mappel::TrustRegionMaximizer< Model >::TrustRegionMaximizer (Model & model, int max_iterations = DEFAULT_ITERATIONS) [inline]`

Definition at line 392 of file estimator.h.

8.53.5 Member Function Documentation

8.53.5.1 `template<class Model> bool mappel::IterativeMaximizer< Model >::backtrack (MaximizerData & data) [protected], [inherited]`

Definition at line 647 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_`, `iteration()`, `mappel::IterativeMaximizer< Model >::MaximizerData::restore_stencil()`, `mappel::methods::objective::rllh()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::save_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::IterativeMaximizer< Model >::MaximizerData::set_stencil()`, `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::step`.

Referenced by `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.53.5.2 `template<class Model> VecT mappel::TrustRegionMaximizer< Model >::bound_step (const VecT & step_hat, const VecT & D, const VecT & theta, const VecT & lbound, const VecT & ubound) [protected]`

This is $\alpha[d]$ from Coleman and Li

Definition at line 1301 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::alpha`, and `mappel::IterativeMaximizer< Model >::MaximizerData::step`.

8.53.5.3 `template<class Model> void mappel::IterativeMaximizer< Model >::clear_stats () [virtual], [inherited]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 625 of file estimator_impl.h.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::mtx`, `mappel::IterativeMaximizer< Model >::total_backtracks`, `mappel::IterativeMaximizer< Model >::total_der_evals`, `mappel::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::IterativeMaximizer< Model >::total_`, `iterations`.

8.53.5.4 `template<class Model > void mappel::TrustRegionMaximizer< Model >::compute_bound_scaling_vec (const VecT & theta, const VecT & g, const VecT & lbound, const VecT & ubound, VecT & v, VecT & Jv) [static], [protected]`

The vector used for bound constrained TR scaling.

This v is from Coleman&Li (1996). It represents a scaling factor for bound constrained problems. For unconstrained problems $v = \text{sgn}(\text{grad})$;

In all cases

Definition at line 1265 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::lbound`, `mappel::sgn()`, `mappel::IterativeMaximizer< Model >::MaximizerData::theta()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::ubound`.

8.53.5.5 `template<class Model > VecT mappel::TrustRegionMaximizer< Model >::compute_cauchy_point (const VecT & g, const MatT & H, double delta) [static], [protected]`

Definition at line 1342 of file estimator_impl.h.

8.53.5.6 `template<class Model > VecT mappel::TrustRegionMaximizer< Model >::compute_D_scale (const VecT & oldDscale, const VecT & grad2) [static], [protected]`

This works for either minimization or maximization. `sign(grad2)` is not important

Definition at line 1226 of file estimator_impl.h.

8.53.5.7 `template<class Model > void mappel::Estimator< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl) [protected], [virtual], [inherited]`

Default base class implementation computes `rllh` and `obsl` separately from stencil This should be overridden by [Estimator](#) subclasses that already have access to this information.

Definition at line 183 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::model`, and `mappel::methods::observed_information()`.

8.53.5.8 `template<class Model > StencilT< Model > mappel::IterativeMaximizer< Model >::compute_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [protected], [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 753 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`.

8.53.5.9 `template<class Model > StencilT< Model > mappel::IterativeMaximizer< Model >::compute_estimate_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamVecT< Model > & sequence, VecT & sequence_rllh)` [protected],[virtual],[inherited]

Estimators that produce a sequence of results (e.g. IterativeEstimators) can override this dummy debug implementation.

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 768 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence_rllh\(\)](#), [mappel::IterativeMaximizer< Model >::max_backtracks](#), [mappel::IterativeMaximizer< Model >::max_iterations](#), [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#).

8.53.5.10 `template<class Model > double mappel::TrustRegionMaximizer< Model >::compute_initial_trust_radius (const VecT & ghat)` [inline],[static],[protected]

Works for minimization or maximization. Independent of sign or grad

Definition at line 1239 of file estimator_impl.h.

8.53.5.11 `template<class Model > void mappel::IterativeMaximizer< Model >::compute_profile_estimate (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, const IdxVecT & fixed_parameters, ParamT< Model > & theta_est, double & rllh)` [protected],[virtual],[inherited]

Reimplemented from [mappel::Estimator< Model >](#).

Definition at line 797 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

8.53.5.12 `template<class Model > bool mappel::IterativeMaximizer< Model >::convergence_test (MaximizerData & data)` [protected],[inherited]

Definition at line 731 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::IterativeMaximizer< Model >::MaximizerData::free_parameters](#), [mappel::IterativeMaximizer< Model >::MaximizerData::grad](#), [mappel::IterativeMaximizer< Model >::MaximizerData::num_fixed_parameters](#), [mappel::ThreadedEstimator< Model >::record_exit_code\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#).

8.53.5.13 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im) [inherited]`

Definition at line 64 of file estimator_impl.h.

References mappel::Estimator< Model >::model.

Referenced by mappel::Estimator< Model >::estimate_max(), and mappel::Estimator< Model >::~Estimator().

8.53.5.14 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init) [inherited]`

Definition at line 82 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::methods::objective::rllh().

8.53.5.15 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, double & rllh) [inherited]`

Definition at line 73 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::Estimator< Model >::model.

8.53.5.16 `template<class Model > StencilT< Model > mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, double & rllh) [inherited]`

Definition at line 90 of file estimator_impl.h.

References mappel::Estimator< Model >::compute_estimate(), mappel::Estimator< Model >::Error, mappel::Estimator< Model >::model, mappel::print_image(), mappel::Estimator< Model >::record_exit_code(), and mappel::Estimator< Model >::record_walltime().

8.53.5.17 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 119 of file estimator_impl.h.

References mappel::Estimator< Model >::estimate_max(), and mappel::Estimator< Model >::model.

8.53.5.18 `template<class Model > void mappel::Estimator< Model >::estimate_max (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta, double & rllh, MatT & obsl) [inherited]`

Definition at line 128 of file estimator_impl.h.

References mappel::Estimator< Model >::compute_estimate(), mappel::Estimator< Model >::Error, mappel::print_image(), mappel::Estimator< Model >::record_exit_code(), mappel::Estimator< Model >::record_walltime(), and mappel::methods::objective::rllh().

8.53.5.19 `template<class Model> void mappel::Estimator< Model >::estimate_max_debug (const ModelDataT< Model > & im, const ParamT< Model > & theta_init, ParamT< Model > & theta_est, double & rllh, MatT & obsl, MatT & sequence, VecT & sequence_rllh) [inherited]`

Definition at line 158 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::model`, `mappel::methods::observed_information()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::rllh()`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.53.5.20 `template<class Model> void mappel::Estimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im_stack, ParamVecT< Model > & theta_est_stack, VecT & rllh_stack, CubeT & obsl_stack) [inherited]`

Definition at line 170 of file estimator_impl.h.

References `mappel::Estimator< Model >::estimate_max_stack()`, and `mappel::Estimator< Model >::model`.

8.53.5.21 `template<class Model> void mappel::ThreadedEstimator< Model >::estimate_max_stack (const ModelDataStackT< Model > & im, const ParamVecT< Model > & theta_init, ParamVecT< Model > & theta, VecT & rllh, CubeT & obsl) [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 285 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, `mappel::Estimator< Model >::record_walltime()`, and `mappel::methods::objective::openmp::rllh_stack()`.

8.53.5.22 `template<class Model> void mappel::ThreadedEstimator< Model >::estimate_profile_stack (const ModelDataT< Model > & data, const IdxVecT & fixed_parameters, const MatT & values, const ParamVecT< Model > & theta_init, VecT & profile_likelihood, ParamVecT< Model > & profile_parameters) [virtual], [inherited]`

Implements [mappel::Estimator< Model >](#).

Definition at line 323 of file estimator_impl.h.

References `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::Error`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::print_image()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::record_walltime()`.

8.53.5.23 `template<class Model> StatsT mappel::IterativeMaximizer< Model >::get_debug_stats () [virtual], [inherited]`

Reimplemented from [mappel::ThreadedEstimator< Model >](#).

Definition at line 609 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`s, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::IterativeMaximizer< Model >::last_backtrack_idx`s.

8.53.5.24 `template<class Model > Model & mappel::Estimator< Model >::get_model ()` [inherited]

Definition at line 50 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.53.5.25 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_stats ()` [virtual], [inherited]

Reimplemented from `mappel::ThreadedEstimator< Model >`.

Definition at line 588 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::IterativeMaximizer< Model >::max_iterations`, `mappel::ThreadedEstimator< Model >::mtx`, `mappel::Estimator< Model >::num_estimations`, `mappel::IterativeMaximizer< Model >::total_backtracks`, `mappel::IterativeMaximizer< Model >::total_der_evals`, `mappel::IterativeMaximizer< Model >::total_fun_evals`, and `mappel::IterativeMaximizer< Model >::total_iterations`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::get_debug_stats()`.

8.53.5.26 `template<class Model > void mappel::IterativeMaximizer< Model >::local_maximize (const ModelDataT< Model > & im, const StencilT< Model > & theta_init, StencilT< Model > & stencil, double & rllh)` [inherited]

Perform a local maximization to finish off a simulated annealing run.

Definition at line 811 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, and `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`.

8.53.5.27 `template<class Model > virtual void mappel::IterativeMaximizer< Model >::maximize (MaximizerData & data)` [protected], [pure virtual], [inherited]

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, and `mappel::IterativeMaximizer< Model >::local_maximize()`.

8.53.5.28 `template<class Model > void mappel::TrustRegionMaximizer< Model >::maximize (MaximizerData & data)` [protected]

Definition at line 1059 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::methods::objective::hessian()`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::methods::objective::rllh()`.

8.53.5.29 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_backtracks ()` [inherited]

8.53.5.30 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_der_evals ()` [inherited]

8.53.5.31 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_fun_evals ()` [inherited]

8.53.5.32 `template<class Model > double mappel::IterativeMaximizer< Model >::mean_iterations ()` [inherited]

8.53.5.33 `template<class Model > std::string mappel::TrustRegionMaximizer< Model >::name () const` [inline], [virtual]

Implements [mappel::Estimator< Model >](#).

Definition at line 395 of file estimator.h.

References [mappel::methods::objective::grad\(\)](#), and [mappel::methods::objective::grad2\(\)](#).

8.53.5.34 `template<class Model > double mappel::TrustRegionMaximizer< Model >::quadratic_model_value (const VecT & step, const VecT & grad, const MatT & hess)` [inline], [static], [protected]

Quadratic model value at given step Compute a quadratic model.

Definition at line 1251 of file estimator_impl.h.

8.53.5.35 `template<class Model > void mappel::ThreadedEstimator< Model >::record_exit_code (ExitCode code)` [protected], [virtual], [inherited]

Implements [mappel::Estimator< Model >](#).

Definition at line 386 of file estimator_impl.h.

References [mappel::Estimator< Model >::exit_counts](#), [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::mtx](#), and [mappel::methods::objective::rllh\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::convergence_test\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::HeuristicEstimator< Model >::get_debug_stats\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

8.53.5.36 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const MaximizerData & data)` [protected], [inherited]

Definition at line 636 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::MaximizerData::get_backtrack_idxes\(\)](#), [mappel::IterativeMaximizer< Model >::last_backtrack_idxes](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::IterativeMaximizer< Model >::MaximizerData::nBacktracks](#), [mappel::IterativeMaximizer< Model >::MaximizerData::nIterations](#), [mappel::IterativeMaximizer< Model >::MaximizerData::save_seq](#), [mappel::IterativeMaximizer< Model >::total_backtracks](#), [mappel::IterativeMaximizer< Model >::total_der_evals](#), [mappel::IterativeMaximizer< Model >::total_fun_evals](#), and [mappel::IterativeMaximizer< Model >::total_iterations](#).

Referenced by [mappel::IterativeMaximizer< Model >::compute_estimate\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), and [mappel::IterativeMaximizer< Model >::compute_profile_estimate\(\)](#).

8.53.5.37 `template<class Model > void mappel::Estimator< Model >::record_walltime (ClockT::time_point start_walltime, int nimages)` `[protected]`, `[inherited]`

Definition at line 266 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::num_estimations`, and `mappel::Estimator< Model >::total_walltime`.

Referenced by `mappel::Estimator< Model >::estimate_max()`, `mappel::Estimator< Model >::estimate_max_↵ debug()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, and `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`.

8.53.5.38 `template<class Model > void mappel::Estimator< Model >::set_model (Model & new_model)` `[inherited]`

Definition at line 54 of file `estimator_impl.h`.

References `mappel::Estimator< Model >::model`.

Referenced by `mappel::Estimator< Model >::~~Estimator()`.

8.53.5.39 `template<class Model > VecT mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton (const VecT & g, const MatT & H, double delta, double lambda_lb, double lambda_ub, double epsilon)` `[static]`, `[protected]`

Definition at line 1451 of file `estimator_impl.h`.

References `mappel::cholesky()`, `mappel::cholesky_convert_lower_triangular()`, `mappel::cholesky_solve()`, `mappel::↵ IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, and `mappel::IterativeMaximizer< Model >::lambda_min`.

8.53.5.40 `template<class Model > VecT mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem (const VecT & g, const MatT & H, double delta, double epsilon)` `[static]`, `[protected]`

Exactly solver the TR subproblem 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 1359 of file `estimator_impl.h`.

References `mappel::cholesky()`, `mappel::cholesky_solve()`, `mappel::IterativeMaximizer< Model >::delta`, and `mappel::↵ IterativeMaximizer< Model >::lambda_min`.

8.53.6 Member Data Documentation

8.53.6.1 `template<class Model > double mappel::IterativeMaximizer< Model >::alpha = 1e-4` `[protected]`, `[inherited]`

Definition at line 245 of file `estimator.h`.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, and `mappel::TrustRegionMaximizer< Model >↵ ::bound_step()`.

8.53.6.2 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::boundary_stepback_min_kappa = 1.0 - 1.0e-5 [static]`

Definition at line 390 of file estimator.h.

8.53.6.3 `template<class Model > double mappel::IterativeMaximizer< Model >::delta = 1e-6 [protected], [inherited]`

Definition at line 242 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`, and `mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem()`.

8.53.6.4 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::delta_decrease = 0.25 [static]`

Definition at line 383 of file estimator.h.

8.53.6.5 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::delta_decrease_min = 0.125 [static]`

Definition at line 382 of file estimator.h.

8.53.6.6 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::delta_increase = 2 [static]`

Definition at line 384 of file estimator.h.

8.53.6.7 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::delta_init_max = 1.0e3 [static]`

Definition at line 389 of file estimator.h.

8.53.6.8 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::delta_init_min = 1.0e-3 [static]`

Definition at line 388 of file estimator.h.

8.53.6.9 `template<class Model > double mappel::IterativeMaximizer< Model >::epsilon = 1e-6 [protected], [inherited]`

Definition at line 241 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`.

8.53.6.10 `template<class Model > IdxVecT mappel::Estimator< Model >::exit_counts` [protected],
[inherited]

Definition at line 114 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, and `mappel::ThreadedEstimator< Model >::record_exit_code()`.

8.53.6.11 `template<class Model > double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` [protected],
[inherited]

Definition at line 244 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`, and `mappel::TrustRegionMaximizer< Model >::solve_TR_subproblem()`.

8.53.6.12 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::last_backtrack_idx`
[protected], [inherited]

Definition at line 255 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::get_debug_stats()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.53.6.13 `template<class Model > int mappel::IterativeMaximizer< Model >::max_backtracks = 8` [protected],
[inherited]

Definition at line 246 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, and `mappel::IterativeMaximizer< Model >::get_stats()`.

8.53.6.14 `template<class Model > int mappel::IterativeMaximizer< Model >::max_iterations` [protected],
[inherited]

Definition at line 238 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

8.53.6.15 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::max_scaling = 1.0e5`
[static]

Definition at line 387 of file estimator.h.

8.53.6.16 `template<class Model > int mappel::ThreadedEstimator< Model >::max_threads` [protected],
[inherited]

Definition at line 142 of file estimator.h.

8.53.6.17 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::min_scaling = 1.0e-5`
`[static]`

Definition at line 386 of file estimator.h.

8.53.6.18 `template<class Model > Model& mappel::Estimator< Model >::model` `[protected]`, `[inherited]`

Definition at line 109 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::Estimator< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::Estimator< Model >::compute_estimate_debug()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::Estimator< Model >::compute_profile_estimate()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::Estimator< Model >::estimate_max()`, `mappel::Estimator< Model >::estimate_max_debug()`, `mappel::Estimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::HeuristicEstimator< Model >::get_debug_stats()`, `mappel::Estimator< Model >::get_model()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::Estimator< Model >::set_model()`.

8.53.6.19 `template<class Model > std::mutex mappel::ThreadedEstimator< Model >::mtx` `[protected]`,
`[inherited]`

Definition at line 144 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::record_exit_code()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.53.6.20 `template<class Model > int mappel::Estimator< Model >::num_estimations = 0` `[protected]`,
`[inherited]`

Definition at line 112 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

8.53.6.21 `template<class Model > int mappel::ThreadedEstimator< Model >::num_threads` `[protected]`,
`[inherited]`

Definition at line 143 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, and `mappel::ThreadedEstimator< Model >::get_stats()`.

8.53.6.22 `template<class Model > constexpr int mappel::Estimator< Model >::NumExitCodes = 9` [static],
[inherited]

Definition at line 39 of file estimator.h.

8.53.6.23 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::rho_cauchy_min = 0.1`
[static]

Definition at line 379 of file estimator.h.

8.53.6.24 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::rho_obj_min = 0.25`
[static]

Definition at line 380 of file estimator.h.

8.53.6.25 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::rho_obj_opt = 0.75`
[static]

Definition at line 381 of file estimator.h.

8.53.6.26 `template<class Model > int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected],
[inherited]

Definition at line 250 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.53.6.27 `template<class Model > int mappel::IterativeMaximizer< Model >::total_der_evals = 0` [protected],
[inherited]

Definition at line 252 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.53.6.28 `template<class Model > int mappel::IterativeMaximizer< Model >::total_fun_evals = 0` [protected],
[inherited]

Definition at line 251 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.53.6.29 `template<class Model > int mappel::IterativeMaximizer< Model >::total_iterations = 0` [protected],
[inherited]

Definition at line 249 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::get_stats()`,
and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.53.6.30 `template<class Model > double mappel::Estimator< Model >::total_walltime = 0. [protected],
[inherited]`

Definition at line 113 of file estimator.h.

Referenced by `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::get_stats()`, `mappel::ThreadedEstimator< Model >::get_stats()`, and `mappel::Estimator< Model >::record_walltime()`.

The documentation for this class was generated from the following files:

- [estimator.h](#)
- [estimator_impl.h](#)

9 File Documentation

9.1 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_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"`

9.1.1 Detailed Description

Author

Mark J. Olah (mjo@cs.unm.edu DOT edu)

Date

03-23-2014

9.2 display.h File Reference

```
#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 Vec >`
`std::ostream & mappel::print_vec_row (std::ostream &out, const Vec &vec, const char *header, int header_width, const char *color=NULLptr)`

9.2.1 Detailed Description

Author

Mark J. Olah (mjo@cs.unm.edu DOT edu)

Date

03-23-2014

9.3 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

- class [mappel::Estimator< Model >](#)
- class [mappel::ThreadedEstimator< Model >](#)
- class [mappel::HeuristicEstimator< Model >](#)
- class [mappel::CGaussHeuristicEstimator< Model >](#)
- class [mappel::CGaussMLE< Model >](#)
- class [mappel::SimulatedAnnealingMaximizer< Model >](#)
- class [mappel::IterativeMaximizer< Model >](#)
- class [mappel::IterativeMaximizer< Model >::MaximizerData](#)
- class [mappel::NewtonDiagonalMaximizer< Model >](#)
- class [mappel::NewtonMaximizer< Model >](#)
- class [mappel::QuasiNewtonMaximizer< Model >](#)
- class [mappel::TrustRegionMaximizer< Model >](#)

Namespaces

- [mappel](#)

Typedefs

- typedef std::chrono::high_resolution_clock [ClockT](#)

9.3.1 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

04-01-2014

9.3.2 Typedef Documentation

9.3.2.1 typedef std::chrono::high_resolution_clock ClockT

Definition at line 25 of file estimator.h.

9.4 estimator_impl.h File Reference

```
#include <thread>
#include <cmath>
#include <armadillo>
#include "estimator.h"
#include "Mappel/rng.h"
#include "Mappel/numerical.h"
#include "Mappel/display.h"
```

Namespaces

- [mappel](#)

Functions

- template<class Model >
std::ostream & [mappel::operator<<](#) (std::ostream &out, Estimator< Model > &estimator)

9.4.1 Detailed Description

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

01-15-2014

9.5 estimator_statics.cpp File Reference

```
#include "Mappel/Gauss1DMAP.h"
#include "Mappel/estimator_impl.h"
```

Namespaces

- [mappel](#)

9.5.1 Detailed Description

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

01-15-2014

9.6 Gauss1DMAP.cpp File Reference

The class definition and template Specializations for Gauss1DMAP.

```
#include "Mappel/Gauss1DMAP.h"
```

Namespaces

- [mappel](#)

9.6.1 Detailed Description

The class definition and template Specializations for Gauss1DMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.7 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::Gauss1DMP](#)

A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.

Namespaces

- [mappel](#)

9.7.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DMP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.8 Gauss1DMLE.cpp File Reference

The class definition and template Specializations for Gauss1DMLE.

```
#include "Mappel/Gauss1DMLE.h"
```

Namespaces

- [mappel](#)

9.8.1 Detailed Description

The class definition and template Specializations for Gauss1DMLE.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.9 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](#)

9.9.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DMLE.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.10 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`)

9.10.1 Detailed Description

The class definition and template Specializations for Gauss1DModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.11 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`

9.11.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.12 Gauss1DsMAP.cpp File Reference

The class definition and template Specializations for Gauss1DsMAP.

```
#include "Mappel/Gauss1DsMAP.h"
```

Namespaces

- [mappel](#)

9.12.1 Detailed Description

The class definition and template Specializations for Gauss1DsMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017

9.13 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](#)

9.13.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DsMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.14 Gauss1DsMLE.cpp File Reference

The class definition and template Specializations for Gauss1DsMLE.

```
#include "Mappel/Gauss1DsMLE.h"
```

Namespaces

- [mappel](#)

9.14.1 Detailed Description

The class definition and template Specializations for Gauss1DsMLE.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017

9.15 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](#)

9.15.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DsMLE.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.16 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)`

9.16.1 Detailed Description

The class definition and template Specializations for Gauss1DsModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.17 Gauss1DsModel.h File Reference

The class declaration and inline and templated functions for Gauss1DsModel.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/ImageFormat1DBase.h"
#include "Mappel/MCMCAAdaptor1Ds.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](#)

9.17.1 Detailed Description

The class declaration and inline and templated functions for Gauss1DsModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.18 Gauss2DMAP.cpp File Reference

The class definition and template Specializations for Gauss2DMAP.

```
#include "Mappel/Gauss2DMAP.h"
```

Namespaces

- [mappel](#)

9.18.1 Detailed Description

The class definition and template Specializations for Gauss2DMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.19 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](#)

9.19.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.20 Gauss2DMLE.cpp File Reference

The class definition and template Specializations for Gauss2DMLE.

```
#include "Mappel/Gauss2DMLE.h"
```

Namespaces

- [mappel](#)

9.20.1 Detailed Description

The class definition and template Specializations for Gauss2DMLE.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.21 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](#)

9.21.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DMLE.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.22 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)`

9.22.1 Detailed Description

The class definition and template Specializations for Gauss2DModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.23 Gauss2DModel.h File Reference

The class declaration and inline and templated functions for Gauss2DModel.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/ImageFormat2DBase.h"
#include "Mappel/MCMCAaptor2D.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)

9.23.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.24 Gauss2DsMAP.cpp File Reference

The class definition and template Specializations for Gauss2DsMAP.

```
#include "Mappel/Gauss2DsMAP.h"
```

Namespaces

- [mappel](#)

9.24.1 Detailed Description

The class definition and template Specializations for Gauss2DsMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.25 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](#)

9.25.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.26 Gauss2DsMLE.cpp File Reference

The class definition and template Specializations for Gauss2DsMLE.

```
#include "Mappel/Gauss2DsMLE.h"
```

Namespaces

- [mappel](#)

9.26.1 Detailed Description

The class definition and template Specializations for Gauss2DsMLE.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.27 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](#)

9.27.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsMLE.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.28 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)`

9.28.1 Detailed Description

The class definition and template Specializations for Gauss2DsModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.29 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)

9.29.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.30 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](#)

9.30.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsxyMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017

9.31 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)`

9.31.1 Detailed Description

The class declaration and inline and templated functions for Gauss2DsxModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.32 ImageFormat1DBase.cpp File Reference

The class definition and template Specializations for ImageFormat1DBase.

```
#include "Mappel/ImageFormat1DBase.h"
```

Namespaces

- [mappel](#)

9.32.1 Detailed Description

The class definition and template Specializations for ImageFormat1DBase.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.33 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)`

9.33.1 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

9.34 ImageFormat2DBase.cpp File Reference

The class definition and template Specializations for ImageFormat2DBase.

```
#include "Mappel/ImageFormat2DBase.h"
```

Namespaces

- [mappel](#)

9.34.1 Detailed Description

The class definition and template Specializations for ImageFormat2DBase.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.35 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)

9.35.1 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

9.36 Install.md File Reference

9.37 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)`

9.37.1 Detailed Description

Class declaration and inline and templated functions for MAPEstimator.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017-2019

9.38 mcmc.cpp File Reference

```
#include "Mappel/util.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`)

9.39 mcmc.h File Reference

Templated MCMC methods for posterior estimation.

```
#include <cmath>
#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 Model >`
`void mappel::mcmc::sample_posterior` (`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` (`Model &model`, `const ModelDataT< Model > &im`, `const StencilT< Model > &theta_init`, `MatT &sample`, `VecT &sample_rllh`, `MatT &candidate`, `VecT &candidate_rllh`)

9.39.1 Detailed Description

Templated MCMC methods for posterior estimation.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2015-2019

9.40 MCMCAdaptor1D.cpp File Reference

The class definition and template Specializations for MCMCAdaptor1D.

```
#include "Mappel/MCMCAdaptor1D.h"
```

Namespaces

- [mappel](#)

9.40.1 Detailed Description

The class definition and template Specializations for MCMCAdaptor1D.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018

9.41 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](#)

9.41.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptor1D.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018-2019

9.42 MCMCAdaptor1Ds.cpp File Reference

The class definition and template Specializations for MCMCAdaptor1Ds.

```
#include "Mappel/MCMCAdaptor1Ds.h"
```

Namespaces

- [mappel](#)

9.42.1 Detailed Description

The class definition and template Specializations for MCMCAdaptor1Ds.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018

9.43 MCMCAdaptor1Ds.h File Reference

The class declaration and inline and templated functions for MCMCAdaptor1Ds.

```
#include "Mappel/MCMCAdaptor1D.h"
```

Classes

- class [mappel::MCMCAdaptor1Ds](#)

Namespaces

- [mappel](#)

9.43.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptor1Ds.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018-2019

9.44 MCMCAdaptor2D.cpp File Reference

The class definition and template Specializations for MCMCAdaptor2D.

```
#include "Mappel/MCMCAdaptor2D.h"
```

Namespaces

- [mappel](#)

9.44.1 Detailed Description

The class definition and template Specializations for MCMCAdaptor2D.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018

9.45 MCMCAdaptor2D.h File Reference

The class declaration and inline and templated functions for MCMCAdaptor2D.

```
#include "Mappel/MCMCAdaptor1D.h"
```

Classes

- class [mappel::MCMCAdaptor2D](#)

Namespaces

- [mappel](#)

9.45.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptor2D.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018-2019

9.46 MCMCAdaptor2Ds.cpp File Reference

The class definition and template Specializations for MCMCAdaptor2Ds.

```
#include "Mappel/MCMCAdaptor2Ds.h"
```

Namespaces

- [mappel](#)

9.46.1 Detailed Description

The class definition and template Specializations for MCMCAdaptor2Ds.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018

9.47 MCMCAdaptor2Ds.h File Reference

The class declaration and inline and templated functions for MCMCAdaptor2Ds.

```
#include "Mappel/MCMCAdaptor2D.h"
```

Classes

- class [mappel::MCMCAdaptor2Ds](#)

Namespaces

- [mappel](#)

9.47.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptor2Ds.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018-2019

9.48 MCMCAdaptorBase.cpp File Reference

The class definition and template Specializations for MCMCAdaptorBase.

```
#include "Mappel/MCMCAdaptorBase.h"
```

Namespaces

- [mappel](#)

9.48.1 Detailed Description

The class definition and template Specializations for MCMCAdaptorBase.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2018

9.49 MCMCAdaptorBase.h File Reference

The class declaration and inline and templated functions for MCMCAdaptorBase.

```
#include "Mappel/util.h"
```

Classes

- class [mappel::MCMCAdaptorBase](#)

Namespaces

- [mappel](#)

9.49.1 Detailed Description

The class declaration and inline and templated functions for MCMCAdaptorBase.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018

9.50 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)`

9.50.1 Detailed Description

Class declaration and inline and templated functions for MLEstimator.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017

9.51 model_methods.h File Reference

```
#include "Mappel/OMPExcceptionCatcher/OMPExcceptionCatcher.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 (Model &model, const ParamT< Model > &theta)`
- `template<class Model, class rng_t >`
`ModelDataT< Model > mappel::methods::simulate_image (Model &model, const ParamT< Model > &theta, rng_t &rng)`
- `template<class Model >`
`ModelDataT< Model > mappel::methods::simulate_image (Model &model, const StencilT< Model > &s)`
- `template<class Model >`
`ModelDataT< Model > mappel::methods::simulate_image_from_model (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_mode)
- `template<class Model >`
`MatT mappel::methods::observed_information` (const Model &model, const ModelDataT< Model > &data, const StencilT< Model > &theta_mode)
- `template<class Model >`
`StencilT< Model > mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method)
- `template<class Model >`
`StencilT< Model > mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, double &rllh)
- `template<class Model >`
`void mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsl)
- `template<class Model >`
`void mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsl, StatsT &stats)
- `template<class Model >`
`void mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsl)
- `template<class Model >`
`void mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsl, StatsT &stats)
- `template<class Model >`
`MatT mappel::methods::estimate_mcmc_sample` (Model &model, const ModelDataT< Model > &data, IdxT Nsample=1000, IdxT Nburnin=100, IdxT thin=0)
- `template<class Model >`
`MatT mappel::methods::estimate_mcmc_sample` (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample=1000, IdxT Nburnin=100, IdxT thin=0)
- `template<class Model >`
`void mappel::methods::estimate_mcmc_sample` (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample, IdxT Nburnin, IdxT thin, MatT &sample, VecT &sample_rllh)
- `template<class Model >`
`void mappel::methods::estimate_mcmc_posterior` (Model &model, const ModelDataT< Model > &data, IdxT Nsample, IdxT Nburnin, IdxT thin, ParamT< Model > &posterior_mean, MatT &posterior_cov)
- `template<class Model >`
`void mappel::methods::estimate_mcmc_posterior` (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample, IdxT Nburnin, IdxT thin, ParamT< Model > &posterior_mean, MatT &posterior_cov)
- `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 ParamT< Model > &theta_est, MatT &obsl, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)
- `template<class Model >`
`void mappel::methods::error_bounds_posterior_credible` (const Model &model, const MatT &sample, double confidence, ParamT< Model > &theta_mean, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)

- `template<class Model >`
`void mappel::methods::debug::estimate_max_debug (Model &model, const ModelDataT< Model > &data, const std::string &method, ParamT< Model > &theta_est, double &rllh, MatT &obsI, MatT &sequence, VecT &sequence_rllh, StatsT &stats)`
- `template<class Model >`
`void mappel::methods::debug::estimate_max_debug (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, ParamT< Model > &theta_est, double &rllh, MatT &obsI, MatT &sequence, VecT &sequence_rllh, StatsT &stats)`
- `template<class Model >`
`void mappel::methods::debug::estimate_mcmc_sample_debug (Model &model, const ModelDataT< Model > &data, IdxT Nsample, MatT &sample, VecT &sample_rllh, MatT &candidates, VecT &candidates_rllh)`
- `template<class Model >`
`void mappel::methods::debug::estimate_mcmc_sample_debug (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample, MatT &sample, VecT &sample_rllh, MatT &candidates, VecT &candidates_rllh)`

9.52 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 (Model &model, const ParamT< Model > &theta)`
- `template<class Model, class RngT >`
`ModelDataT< Model > mappel::methods::simulate_image (Model &model, const ParamT< Model > &theta, RngT &rng)`
- `template<class Model >`
`ModelDataT< Model > mappel::methods::simulate_image (Model &model, const StencilT< Model > &s)`
- `template<class Model >`
`ModelDataT< Model > mappel::methods::simulate_image_from_model (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_mode)
- `template<class Model >`
`MatT mappel::methods::observed_information` (const Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_mode)
- `template<class Model >`
`StencilT< Model > mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method)
- `template<class Model >`
`StencilT< Model > mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, double &rllh)
- `template<class Model >`
`void mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsI)
- `template<class Model >`
`void mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsI, StatsT &stats)
- `template<class Model >`
`void mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsI)
- `template<class Model >`
`void mappel::methods::estimate_max` (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, ParamT< Model > &theta_max, double &theta_max_llh, MatT &obsI, StatsT &stats)
- `template<class Model >`
`MatT mappel::methods::estimate_mcmc_sample` (Model &model, const ModelDataT< Model > &data, IdxT Nsample=1000, IdxT Nburnin=100, IdxT thin=0)
- `template<class Model >`
`MatT mappel::methods::estimate_mcmc_sample` (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample=1000, IdxT Nburnin=100, IdxT thin=0)
- `template<class Model >`
`void mappel::methods::estimate_mcmc_sample` (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample, IdxT Nburnin, IdxT thin, MatT &sample, VecT &sample_rllh)

- `template<class Model >`
`void mappel::methods::estimate_mcmc_posterior` (Model &model, const ModelDataT< Model > &data, IdxT Nsample, IdxT Nburnin, IdxT thin, ParamT< Model > &posterior_mean, MatT &posterior_cov)
- `template<class Model >`
`void mappel::methods::estimate_mcmc_posterior` (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample, IdxT Nburnin, IdxT thin, ParamT< Model > &posterior_mean, MatT &posterior_cov)
- `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 ParamT< Model > &theta_est, MatT &obsI, double confidence, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)
- `template<class Model >`
`void mappel::methods::error_bounds_posterior_credible` (const Model &model, const MatT &sample, double confidence, ParamT< Model > &theta_mean, ParamT< Model > &theta_lb, ParamT< Model > &theta_ub)
- `template<class Model >`
`void mappel::methods::debug::estimate_max_debug` (Model &model, const ModelDataT< Model > &data, const std::string &method, ParamT< Model > &theta_est, double &rllh, MatT &obsI, MatT &sequence, VecT &sequence_rllh, StatsT &stats)
- `template<class Model >`
`void mappel::methods::debug::estimate_max_debug` (Model &model, const ModelDataT< Model > &data, const std::string &method, const ParamT< Model > &theta_init, ParamT< Model > &theta_est, double &rllh, MatT &obsI, MatT &sequence, VecT &sequence_rllh, StatsT &stats)
- `template<class Model >`
`void mappel::methods::debug::estimate_mcmc_sample_debug` (Model &model, const ModelDataT< Model > &data, IdxT Nsample, MatT &sample, VecT &sample_rllh, MatT &candidates, VecT &candidates_rllh)
- `template<class Model >`
`void mappel::methods::debug::estimate_mcmc_sample_debug` (Model &model, const ModelDataT< Model > &data, const ParamT< Model > &theta_init, IdxT Nsample, MatT &sample, VecT &sample_rllh, MatT &candidates, VecT &candidates_rllh)

9.53 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)

9.53.1 Detailed Description

Numerical matrix operations.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

05-2015

9.54 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)

9.54.1 Detailed Description

Numerical matrix operations.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

05-22-2015

9.55 OMPEXceptionCatcher.h File Reference

A lightweight class for managing C++ exception handling strategies for openMP methods.

```
#include <exception>
#include <mutex>
#include <functional>
#include <cstdint>
```

Classes

- class [omp_exception_catcher::impl_::OMPEXceptionCatcher< IntType >](#)

Namespaces

- [omp_exception_catcher](#)
- [omp_exception_catcher::impl_](#)

Typedefs

- using `omp_exception_catcher::OMPEExceptionCatcher` = `impl_::OMPEExceptionCatcher< uint32_t >`

9.55.1 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 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 `OMPEExceptionCatcher::Strategies` enum.

Strategy's : `OMPEExceptionCatcher::Strategies::DoNotTry` – Don't even try, this is a null op to completely disable this class's effect. `OMPEExceptionCatcher::Strategies::Continue` – Catch exceptions and keep going `OMPEExceptionCatcher::Strategies::Abort` – Catch exceptions and abort `OMPEExceptionCatcher::Strategies::RethrowFirst` – Re-throws first exception thrown by any thread

Example useage: `OMPEExceptionCatcher catcher(OMPEExceptionCatcher<>::Strategies::Continue); #pragma omp parallel for for(int n=0; n < N; n++) catcher.run([&]{ my_ouput(n)=do_my calulations(args(n)); } catcher.rethrow()); //Required only if you ever might use RethrowFirst strategy`

9.56 openmp_methods.h File Reference

Namespaces for OpenMP parallelized versions of the `mappel::model` namespace functions (external methods)

```
#include <omp.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 (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 (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 (Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_max_stack, VecT &theta_max_rllh, CubeT &obsI_stack)`
- `template<class Model >`
`void mappel::methods::openmp::estimate_max_stack (Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_max_stack, VecT &theta_max_rllh, CubeT &obsI_stack, StatsT &stats)`
- `template<class Model >`
`void mappel::methods::openmp::estimate_max_stack (Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_init_stack, ParamVecT< Model > &theta_max_stack, VecT &theta_max_rllh, CubeT &obsI_stack)`
- `template<class Model >`
`void mappel::methods::openmp::estimate_max_stack (Model &model, const ModelDataStackT< Model > &data_stack, const std::string &method, ParamVecT< Model > &theta_init_stack, ParamVecT< Model > &theta_max_stack, VecT &theta_max_rllh, CubeT &obsI_stack, StatsT &stats)`
- `template<class Model >`
`void mappel::methods::openmp::estimate_profile_likelihood (Model &model, const ModelDataT< Model > &data, const IdxVecT &fixed_parameters, const MatT &fixed_values, const std::string &method, VecT &profile_likelihood, ParamVecT< Model > &profile_parameters)`
- `template<class Model >`
`void mappel::methods::openmp::estimate_profile_likelihood (Model &model, const ModelDataT< Model > &data, const IdxVecT &fixed_parameters, const MatT &fixed_values, const std::string &method, VecT &profile_likelihood, ParamVecT< Model > &profile_parameters, StatsT &stats)`
- `template<class Model >`
`void mappel::methods::openmp::estimate_profile_likelihood (Model &model, const ModelDataT< Model > &data, const IdxVecT &fixed_parameters, const MatT &fixed_values, const std::string &method, const ParamVecT< Model > &theta_init, VecT &profile_likelihood, ParamVecT< Model > &profile_parameters)`
- `template<class Model >`
`void mappel::methods::openmp::estimate_profile_likelihood (Model &model, const ModelDataT< Model > &data, const IdxVecT &fixed_parameters, const MatT &fixed_values, const std::string &method, const ParamVecT< Model > &theta_init, VecT &profile_likelihood, ParamVecT< Model > &profile_parameters, StatsT &stats)`

- `template<class Model >`
`void mappel::methods::openmp::estimate_mcmc_sample_stack` (Model &model, const ModelDataStackT< Model > &data_stack, const ParamVecT< Model > &theta_init_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, CubeT &sample_stack, MatT &sample_rllh_stack)
- `template<class Model >`
`void mappel::methods::openmp::estimate_mcmc_sample_stack` (Model &model, const ModelDataStackT< Model > &data_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, CubeT &sample, MatT &sample_rllh)
- `template<class Model >`
`void mappel::methods::openmp::estimate_mcmc_posterior_stack` (Model &model, const ModelDataStackT< Model > &data_stack, const ParamVecT< Model > &theta_init_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, MatT &theta_mean_stack, CubeT &theta_cov_stack)
- `template<class Model >`
`void mappel::methods::openmp::estimate_mcmc_posterior_stack` (Model &model, const ModelDataStackT< Model > &data_stack, IdxT Nsamples, IdxT Nburnin, IdxT thin, MatT &theta_mean_stack, CubeT &theta_cov←_stack)
- `template<class Model >`
`void mappel::methods::openmp::error_bounds_expected_stack` (const Model &model, const MatT &theta_est←_stack, double confidence, MatT &theta_lb_stack, MatT &theta_ub_stack)
- `template<class Model >`
`void mappel::methods::openmp::error_bounds_observed_stack` (const Model &model, const MatT &theta_est←_stack, CubeT &obsl_stack, double confidence, MatT &theta_lb_stack, MatT &theta_ub_stack)
- `template<class Model >`
`void mappel::methods::openmp::error_bounds_posterior_credible_stack` (const Model &model, const CubeT &sample_stack, double confidence, MatT &theta_mean_stack, MatT &theta_lb_stack, MatT &theta_ub_stack)
- `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.

9.56.1 Detailed Description

Namespaces for OpenMP parallelized versions of the mappel::model namespace functions (external methods)

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2013-2019 OpenMP computation for stacked Model operations on vector data.

Design Decisions

- OpenMP vectorized versions are implemented as templated external methods in inline namespaces openmp. This allows easy future replacement with other parallelization mechanisms (CUDA, C++11 threads, etc..). Also allows the vectorized versions to directly overload with the non-vectorized base-versions.
- Because we want to integrate as seamlessly as possible with matlab, we use the armadillo package which stores arrays in column major order.
- Therefore in the *_stack operations, if they are to be parallelized, we want the data stored as a nParms X n matrix, i.e. each column is a parameter matrix. Similarly stacks are size X size X n, so that contiguous images sequences are contiguous in memory. This avoids false sharing.

9.57 PointEmitterModel.cpp File Reference

The class definition and template Specializations for PointEmitterModel.

```
#include <cmath>
#include <algorithm>
#include "Mappel/PointEmitterModel.h"
```

Namespaces

- [mappel](#)

9.57.1 Detailed Description

The class definition and template Specializations for PointEmitterModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

03-13-2014

9.58 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)`

9.58.1 Detailed Description

The class declaration and inline and templated functions for PointEmitterModel.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

03-13-2014 The base class for all point emitter localization models

9.59 PoissonGaussianNoise2DObjective.cpp File Reference

```
#include "Mappel/PoissonGaussianNoise2DObjective.h"
```

Namespaces

- [mappel](#)

9.60 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)`

9.60.1 Detailed Description

The class declaration and inline and templated functions for [PoissonGaussianNoise2DObjective](#).

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

04-2017

9.60.2 Function Documentation

9.60.2.1 `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](#).

Referenced by `mappel::fisher_information_stack()`.

9.60.2.2 `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()`.

Referenced by `mappel::log_likelihood_stack()`.

```
9.60.2.3 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().

```
9.60.2.4 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.

Referenced by mappel::model_grad_stack().

```
9.60.2.5 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.

```
9.60.2.6 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.

Referenced by mappel::model_hessian_stack().

```
9.60.2.7 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.

Referenced by mappel::model_image_stack(), and mappel::simulate_image_stack().

```
9.60.2.8 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().

```
9.60.2.9 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

out	<i>image</i>	An image to populate.
in	<i>theta</i>	The parameter values to us
in, out	<i>rng</i>	An initialized random number generator

Definition at line 78 of file PoissonGaussianNoise2DObjective.h.

References `mappel::generate_poisson()`.

9.60.2.10 `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()`.

9.61 PoissonNoise1DObjective.cpp File Reference

The class definition and template Specializations for PoissonNoise1DObjective.

```
#include "Mappel/PoissonNoise1DObjective.h"
```

Namespaces

- [mappel](#)

9.61.1 Detailed Description

The class definition and template Specializations for PoissonNoise1DObjective.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.62 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< 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)`

9.62.1 Detailed Description

The class declaration and inline and templated functions for `PoissonNoise1DObjective`.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.63 PoissonNoise2DObjective.cpp File Reference

The class definition and template Specializations for `PoissonNoise2DObjective`.

```
#include "Mappel/PoissonNoise2DObjective.h"
```

Namespaces

- [mappel](#)

9.63.1 Detailed Description

The class definition and template Specializations for PoissonNoise2DObjective.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.64 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 phontons 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< 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)`

9.64.1 Detailed Description

The class declaration and inline and templated functions for `PoissonNoise2DObjective`.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2014-2019

9.65 PriorMAP1DObjective.h File Reference

The class declaration and inline and templated functions for PriorMAP1DObjective.

```
#include <armadillo>
#include "Mappel/stencil.h"
```

Classes

- class [mappel::PriorMAP1DObjective](#)
A Mixin class to configure a [Gauss1DModel](#) for MAP estimation (default 1D prior).

Namespaces

- [mappel](#)

9.65.1 Detailed Description

The class declaration and inline and templated functions for PriorMAP1DObjective.

Author

Mark J. Olah (mjo@cs.unm.edu)

Date

03-22-2014

9.66 README.md File Reference**9.67 rng.cpp File Reference**

```
#include "Mappel/rng.h"
```

Namespaces

- [mappel](#)

Variables

- `ParallelRngManagerT` [mappel::rng_manager](#)

9.68 rng.h File Reference

Random number generation usign sfmt.

```
#include <random>
#include "Mappel/util.h"
#include <trng/lcg64_shift.hpp>
#include <ParallelRngManager/ParallelRngManager.h>
```

Namespaces

- [mappel](#)

Typedefs

- using [mappel::ParallelRngGeneratorT](#) = `trng::lcg64_shift`
- using [mappel::ParallelRngManagerT](#) = `parallel_rng::ParallelRngManager< ParallelRngGeneratorT >`
- using [mappel::RngSeedT](#) = `parallel_rng::SeedT`
- using [mappel::UniformDistT](#) = `std::uniform_real_distribution< double >`

Functions

- `template<class RngT >`
`IdxT mappel::generate_poisson_small (RngT &rng, double mu)`
Genrates a single poisson disributed 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)`

9.68.1 Detailed Description

Random number generation usign sfmt.

Author

Mark J. Olah (email mjo@cs.unm DOT edu)

Date

12-12-2013

9.69 stackcomp.h File Reference

Data-parallel versions of core computational functions using OpenMP.

```
#include <omp.h>
#include "Mappel/rng.h"
```

Namespaces

- [mappel](#)

Functions

- `template<class Model >`
`void mappel::sample_prior_stack` (Model &model, typename Model::ParamVecT &theta_stack)
Parallel sampling of the model prior.
- `template<class Model >`
`void mappel::model_image_stack` (const Model &model, const typename Model::ParamVecT &theta_stack, typename Model::ImageStackT &image_stack)
Parallel computation of the model image.
- `template<class Model >`
`void mappel::simulate_image_stack` (const Model &model, const typename Model::ParamVecT &theta_stack, typename Model::ImageStackT &image_stack)
Parallel simulation of images from one or more theta.
- `template<class Model >`
`void mappel::log_likelihood_stack` (const Model &model, const typename Model::ImageT &image, const typename Model::ParamVecT &theta_stack, VecT &llh_stack)
Parallel log_likelihood calculations for a single image.
- `template<class Model >`
`void mappel::log_likelihood_stack` (const Model &model, const typename Model::ImageStackT &image_stack, const typename Model::ParamVecT &theta_stack, VecT &llh_stack)
Parallel log_likelihood calculations for a stack of images.
- `template<class Model >`
`void mappel::model_grad_stack` (const Model &model, const typename Model::ImageStackT &image_stack, const typename Model::ParamVecT &theta_stack, typename Model::ParamVecT &grad_stack)
Parallel model gradient calculations for a stack of images.
- `template<class Model >`
`void mappel::model_hessian_stack` (const Model &model, const typename Model::ImageStackT &image_stack, const typename Model::ParamVecT &theta_stack, CubeT &hessian_stack)
Parallel model Hessian calculations for a stack of images.
- `template<class Model >`
`void mappel::model_positive_hessian_stack` (const Model &model, const typename Model::ImageStackT &image_stack, const typename Model::ParamVecT &theta_stack, CubeT &hessian_stack)
Parallel model positive-definite Hessian approximation calculations for a stack of images.
- `template<class Model >`
`void mappel::cr_lower_bound_stack` (const Model &model, const typename Model::ParamVecT &theta_stack, typename Model::ParamVecT &crlb_stack)
- `template<class Model >`
`void mappel::fisher_information_stack` (const Model &model, const typename Model::ParamVecT &theta_stack, CubeT &fisherI_stack)

9.69.1 Detailed Description

Data-parallel versions of core computational functions using OpenMP.

Author

Mark J. Olah (mjo@cs.unm.edu)

Date

2013-2017 OpenMP computation for stacked Model operations on vector data.

- Design Decisions
- 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.

9.70 stencil.cpp File Reference

The stencils for pixel based computations.

```
#include <sstream>
#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)
- 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)

9.70.1 Detailed Description

The stencils for pixel based computations.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

03-22-2014

9.71 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 &min_val)
- 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 &min_val)
- double `mappel::gaussian_3D_convolution` (int x, int y, int z, const CubeT &data, const VecFieldT &stencils)
- double `mappel::poisson_log_likelihood` (double model_val, double data_val)
- double `mappel::relative_poisson_log_likelihood` (double model_val, double data_val)
- double `mappel::check_lower_bound_hyperparameter` (const char *name, double value, double lower_bound)
- double `mappel::check_positive_hyperparameter` (const char *name, double value, double hyperprior_epsilon=1E-6)
- double `mappel::check_unit_hyperparameter` (const char *name, double value, double hyperprior_epsilon=1E-6)
- double `mappel::log_prior_beta_const` (double beta)
- double `mappel::log_prior_beta2_const` (double beta0, double beta1)
- double `mappel::log_prior_gamma_const` (double kappa, double mean)
- double `mappel::log_prior_pareto_const` (double alpha, double min)
- double `mappel::log_prior_normal_const` (double sigma)
- double `mappel::rllh_beta_prior` (double beta, double v, double max=1., double min=0.)
- double `mappel::rllh_beta2_prior` (double beta0, double beta1, double v, double max=1., double min=0.)
- double `mappel::rllh_gamma_prior` (double kappa, double mean, double v)
- double `mappel::rllh_pareto_prior` (double alpha, double v)
- double `mappel::rllh_normal_prior` (double mu, double sigma)
- double `mappel::beta_prior_grad` (double beta, double v, double max=1., double min=0.)
- double `mappel::beta2_prior_grad` (double beta0, double beta1, double v, double max=1., double min=0.)
- double `mappel::gamma_prior_grad` (double kappa, double mean, double v)
- double `mappel::pareto_prior_grad` (double alpha, double v)
- double `mappel::normal_prior_grad` (double mu, double sigma)
- double `mappel::beta_prior_grad2` (double beta, double v, double max=1., double min=0.)
- double `mappel::beta2_prior_grad2` (double beta0, double beta1, double v, double max=1., double min=0.)
- double `mappel::gamma_prior_grad2` (double kappa, double v)
- double `mappel::pareto_prior_grad2` (double alpha, double v)
- double `mappel::normal_prior_grad` (double sigma)
- double `mappel::normal_quantile_twosided` (double confidence)
- double `mappel::normal_quantile_onesided` (double confidence)
- double `mappel::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)

9.71.1 Detailed Description

The stencils for pixel based computations.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

03-22-2014

9.72 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)

9.73 util.h File Reference

```
#include <cstdint>
#include <cmath>
#include <memory>
#include <utility>
#include <string>
#include <map>
#include <sstream>
#include <armadillo>
#include <BacktraceException/BacktraceException.h>
```

Classes

- struct [mappel::ParameterValueError](#)
Parameter value is not valid.
- struct [mappel::ArrayShapeError](#)
Array is not of the right dimensionality.
- struct [mappel::ArraySizeError](#)
Array is not of the right size.
- struct [mappel::ModelBoundsError](#)
Access outside the model bounds is attempted.
- struct [mappel::NumericalError](#)
Expected numerical condition does not hold.
- struct [mappel::LogicalError](#)
Failure of code or algorithm logic.
- struct [mappel::NotImplementedError](#)
Feature not yet implemented.

Namespaces

- [mappel](#)

Typedefs

- using [mappel::BoolT](#) = uint16_t
- using [mappel::BoolVecT](#) = arma::Col< uint16_t >
- using [mappel::IdxT](#) = arma::uword
- using [mappel::IdxVecT](#) = arma::Col< IdxT >
- using [mappel::IdxMatT](#) = arma::Mat< IdxT >
- using [mappel::VecT](#) = arma::vec
- using [mappel::MatT](#) = arma::mat
- using [mappel::CubeT](#) = arma::cube
- using [mappel::VecFieldT](#) = arma::field< VecT >
- using [mappel::StatsT](#) = std::map< std::string, double >
- using [mappel::StringVecT](#) = std::vector< std::string >
- template<class ModelT, class ModelBaseT >
using [mappel::EnableIfSubclassT](#) = typename std::enable_if< std::is_base_of< ModelBaseT, ModelT >::value, void >::type
- template<class ReturnT, class ModelT, class ModelBaseT >
using [mappel::ReturnIfSubclassT](#) = typename std::enable_if< std::is_base_of< ModelBaseT, ModelT >::value, ReturnT >::type
- template<class Model >
using [mappel::ImageCoordT](#) = typename Model::ImageCoordT
- template<class Model >
using [mappel::ImagePixelT](#) = typename Model::ImagePixelT
- template<class Model >
using [mappel::ParamT](#) = typename Model::ParamT
- template<class Model >
using [mappel::ParamVecT](#) = typename Model::ParamVecT
- template<class Model >
using [mappel::ImageT](#) = typename Model::ImageT
- template<class Model >
using [mappel::ModelDataT](#) = typename Model::ModelDataT
- template<class Model >
using [mappel::StencilT](#) = typename Model::Stencil
- template<class Model >
using [mappel::ImageStackT](#) = typename Model::ImageStackT
- template<class Model >
using [mappel::ModelDataStackT](#) = typename Model::ModelDataStackT
- template<class Model >
using [mappel::StencilVecT](#) = typename Model::StencilVecT
- using [mappel::MappelError](#) = backtrace_exception::BacktraceException

Functions

- void `mappel::enable_all_cpus` ()
- bool `mappel::istarts_with` (const char *s, const char *pattern)
- bool `mappel::istarts_with` (const std::string &str, const char *pattern)
- const char * `mappel::icontains` (const char *s, const char *pattern)
- int `mappel::maxidx` (const VecT &v)
- template<typename T >
int `mappel::sgn` (T val)
sign (signum) function: -1/0/1
- template<typename T >
T `mappel::square` (T x)
- double `mappel::restrict_value_range` (double val, double minval, double maxval)
- template<typename T , typename... Args>
std::unique_ptr< T > `mappel::make_unique` (Args &&...args)
- std::ostream & `mappel::operator<<` (std::ostream &out, const StatsT &stats)

Index

- ~Estimator
 - mappel::Estimator, [106](#)
- Abort
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [687](#)
- alpha
 - mappel::IterativeMaximizer, [541](#)
 - mappel::NewtonDiagonalMaximizer, [668](#)
 - mappel::NewtonMaximizer, [681](#)
 - mappel::QuasiNewtonMaximizer, [733](#)
 - mappel::TrustRegionMaximizer, [788](#)
- anneal
 - mappel::SimulatedAnnealingMaximizer, [739](#)
- aposteriori_objective
 - mappel::methods, [53](#)
- ArrayShapeError
 - mappel::ArrayShapeError, [86](#)
- ArraySizeError
 - mappel::ArraySizeError, [87](#)
- backtrack
 - mappel::IterativeMaximizer, [535](#)
 - mappel::NewtonDiagonalMaximizer, [662](#)
 - mappel::NewtonMaximizer, [675](#)
 - mappel::QuasiNewtonMaximizer, [727](#)
 - mappel::TrustRegionMaximizer, [781](#)
- backtrack_idxes
 - mappel::IterativeMaximizer::MaximizerData, [565](#)
- beta2_prior_grad
 - mappel, [28](#)
- beta2_prior_grad2
 - mappel, [28](#)
- beta_prior_grad
 - mappel, [28](#)
- beta_prior_grad2
 - mappel, [28](#)
- bg
 - mappel::Gauss1DModel::Stencil, [747](#)
 - mappel::Gauss1DsModel::Stencil, [751](#)
 - mappel::Gauss2DModel::Stencil, [766](#)
 - mappel::Gauss2DsModel::Stencil, [760](#)
 - mappel::Gauss2DsxyModel::Stencil, [755](#)
- BoolVecT
 - mappel, [25](#)
- BoolT
 - mappel, [25](#)
- bound_step
 - mappel::TrustRegionMaximizer, [781](#)
- bound_theta
 - mappel::Gauss1DMAP, [118](#)
 - mappel::Gauss1DMLE, [146](#)
- mappel::Gauss1DModel, [173](#)
- mappel::Gauss1DsMAP, [201](#)
- mappel::Gauss1DsMLE, [229](#)
- mappel::Gauss1DsModel, [257](#)
- mappel::Gauss2DMAP, [285](#)
- mappel::Gauss2DMLE, [315](#)
- mappel::Gauss2DModel, [344](#)
- mappel::Gauss2DsMAP, [373](#)
- mappel::Gauss2DsMLE, [405](#)
- mappel::Gauss2DsModel, [437](#)
- mappel::Gauss2DsxyMAP, [467](#)
- mappel::Gauss2DsxyModel, [494](#)
- mappel::MAPEstimator, [548](#)
- mappel::MCMCAdaptor1Ds, [589](#)
- mappel::MCMCAdaptor1D, [571](#)
- mappel::MCMCAdaptor2Ds, [626](#)
- mappel::MCMCAdaptor2D, [607](#)
- mappel::MLEstimator, [646](#)
- mappel::PointEmitterModel, [693](#)
- boundary_stepback_min_kappa
 - mappel::TrustRegionMaximizer, [788](#)
- bounded_theta
 - mappel::Gauss1DMAP, [118](#)
 - mappel::Gauss1DMLE, [146](#)
 - mappel::Gauss1DModel, [173](#)
 - mappel::Gauss1DsMAP, [201](#)
 - mappel::Gauss1DsMLE, [229](#)
 - mappel::Gauss1DsModel, [257](#)
 - mappel::Gauss2DMAP, [285](#)
 - mappel::Gauss2DMLE, [315](#)
 - mappel::Gauss2DModel, [344](#)
 - mappel::Gauss2DsMAP, [373](#)
 - mappel::Gauss2DsMLE, [405](#)
 - mappel::Gauss2DsModel, [437](#)
 - mappel::Gauss2DsxyMAP, [467](#)
 - mappel::Gauss2DsxyModel, [494](#)
 - mappel::MAPEstimator, [548](#)
 - mappel::MCMCAdaptor1Ds, [589](#)
 - mappel::MCMCAdaptor1D, [571](#)
 - mappel::MCMCAdaptor2Ds, [626](#)
 - mappel::MCMCAdaptor2D, [607](#)
 - mappel::MLEstimator, [646](#)
 - mappel::PointEmitterModel, [693](#)
- bounded_theta_stack
 - mappel::Gauss1DMAP, [118](#)
 - mappel::Gauss1DMLE, [146](#)
 - mappel::Gauss1DModel, [174](#)
 - mappel::Gauss1DsMAP, [201](#)
 - mappel::Gauss1DsMLE, [229](#)
 - mappel::Gauss1DsModel, [257](#)
 - mappel::Gauss2DMAP, [285](#)

- mappel::Gauss2DMLE, 315
- mappel::Gauss2DModel, 344
- mappel::Gauss2DsMAP, 374
- mappel::Gauss2DsMLE, 406
- mappel::Gauss2DsModel, 437
- mappel::Gauss2DsxyMAP, 468
- mappel::Gauss2DsxyModel, 494
- mappel::MAPEstimator, 549
- mappel::MCMCAdaptor1Ds, 590
- mappel::MCMCAdaptor1D, 571
- mappel::MCMCAdaptor2Ds, 626
- mappel::MCMCAdaptor2D, 608
- mappel::MLEstimator, 647
- mappel::PointEmitterModel, 693
- bounds_epsilon
 - mappel::Gauss1DMAP, 133
 - mappel::Gauss1DMLE, 161
 - mappel::Gauss1DModel, 188
 - mappel::Gauss1DsMAP, 216
 - mappel::Gauss1DsMLE, 244
 - mappel::Gauss1DsModel, 272
 - mappel::Gauss2DMAP, 300
 - mappel::Gauss2DMLE, 330
 - mappel::Gauss2DModel, 359
 - mappel::Gauss2DsMAP, 391
 - mappel::Gauss2DsMLE, 423
 - mappel::Gauss2DsModel, 454
 - mappel::Gauss2DsxyMAP, 482
 - mappel::Gauss2DsxyModel, 506
 - mappel::MAPEstimator, 557
 - mappel::MCMCAdaptor1Ds, 599
 - mappel::MCMCAdaptor1D, 581
 - mappel::MCMCAdaptor2Ds, 636
 - mappel::MCMCAdaptor2D, 617
 - mappel::MLEstimator, 655
 - mappel::PointEmitterModel, 701
- CGaussHeuristicEstimator
 - mappel::CGaussHeuristicEstimator, 89
- CGaussMLE
 - mappel::CGaussMLE, 97
- cgauss_compute_estimate
 - mappel, 28
- cgauss_compute_estimate_debug
 - mappel, 28, 29
- cgauss_heuristic_compute_estimate
 - mappel, 29
- check_image_shape
 - mappel::Gauss1DMAP, 119
 - mappel::Gauss1DMLE, 146
 - mappel::Gauss1DModel, 174
 - mappel::Gauss1DsMAP, 201
 - mappel::Gauss1DsMLE, 230
 - mappel::Gauss1DsModel, 257
- mappel::Gauss2DMAP, 285
- mappel::Gauss2DMLE, 315
- mappel::Gauss2DModel, 344
- mappel::Gauss2DsMAP, 374
- mappel::Gauss2DsMLE, 406
- mappel::Gauss2DsModel, 438
- mappel::Gauss2DsxyMAP, 468
- mappel::Gauss2DsxyModel, 494
- mappel::ImageFormat1DBase, 523
- mappel::ImageFormat2DBase, 529
- mappel::PoissonNoise1DObjective, 710
- mappel::PoissonNoise2DObjective, 717
- check_lower_bound_hyperparameter
 - mappel, 29
- check_param_shape
 - mappel::Gauss1DMAP, 119
 - mappel::Gauss1DMLE, 147
 - mappel::Gauss1DModel, 174
 - mappel::Gauss1DsMAP, 202
 - mappel::Gauss1DsMLE, 230
 - mappel::Gauss1DsModel, 257
 - mappel::Gauss2DMAP, 285, 286
 - mappel::Gauss2DMLE, 315, 316
 - mappel::Gauss2DModel, 345
 - mappel::Gauss2DsMAP, 374
 - mappel::Gauss2DsMLE, 406
 - mappel::Gauss2DsModel, 438
 - mappel::Gauss2DsxyMAP, 468
 - mappel::Gauss2DsxyModel, 494, 495
 - mappel::MAPEstimator, 549
 - mappel::MCMCAdaptor1Ds, 590
 - mappel::MCMCAdaptor1D, 572
 - mappel::MCMCAdaptor2Ds, 626
 - mappel::MCMCAdaptor2D, 608
 - mappel::MLEstimator, 647
 - mappel::PointEmitterModel, 693
- check_positive_hyperparameter
 - mappel, 29
- check_psf_sigma
 - mappel::Gauss1DMAP, 119
 - mappel::Gauss1DMLE, 147
 - mappel::Gauss1DModel, 174, 175
 - mappel::Gauss1DsMAP, 202
 - mappel::Gauss1DsMLE, 230
 - mappel::Gauss1DsModel, 258
 - mappel::Gauss2DMAP, 286
 - mappel::Gauss2DMLE, 316
 - mappel::Gauss2DModel, 345
 - mappel::Gauss2DsMAP, 374, 375
 - mappel::Gauss2DsMLE, 406, 407
 - mappel::Gauss2DsModel, 438
 - mappel::Gauss2DsxyMAP, 468, 469
 - mappel::Gauss2DsxyModel, 495
 - mappel::MAPEstimator, 549

- mappel::MCMCAdaptor1Ds, 590
- mappel::MCMCAdaptor1D, 572
- mappel::MCMCAdaptor2Ds, 627
- mappel::MCMCAdaptor2D, 608
- mappel::MLEstimator, 647
- mappel::PointEmitterModel, 694
- check_size
 - mappel::Gauss1DModel, 120
 - mappel::Gauss1DModel, 147
 - mappel::Gauss1DModel, 175
 - mappel::Gauss1DsMAP, 202
 - mappel::Gauss1DsMLE, 231
 - mappel::Gauss1DsModel, 258
 - mappel::Gauss2DModel, 286
 - mappel::Gauss2DModel, 316
 - mappel::Gauss2DModel, 345
 - mappel::Gauss2DsMAP, 375
 - mappel::Gauss2DsMLE, 407
 - mappel::Gauss2DsModel, 439
 - mappel::Gauss2DsxyMAP, 469
 - mappel::Gauss2DsxyModel, 495
 - mappel::ImageFormat1DBase, 523
 - mappel::ImageFormat2DBase, 529
 - mappel::PoissonNoise1DObjective, 710
 - mappel::PoissonNoise2DObjective, 718
- check_unit_hyperparameter
 - mappel, 29
- cholesky
 - mappel, 29
- cholesky_convert_full_matrix
 - mappel, 30
- cholesky_convert_lower_triangular
 - mappel, 30
- cholesky_make_negative_definite
 - mappel, 30
- cholesky_make_positive_definite
 - mappel, 30
- cholesky_solve
 - mappel, 30
- clear_stats
 - mappel::CGaussHeuristicEstimator, 89
 - mappel::CGaussMLE, 98
 - mappel::Estimator, 106
 - mappel::HeuristicEstimator, 514
 - mappel::IterativeMaximizer, 535
 - mappel::NewtonDiagonalMaximizer, 662
 - mappel::NewtonMaximizer, 675
 - mappel::QuasiNewtonMaximizer, 727
 - mappel::SimulatedAnnealingMaximizer, 739
 - mappel::ThreadedEstimator, 771
 - mappel::TrustRegionMaximizer, 781
- ClockT
 - estimator.h, 796
- compute_D_scale
 - mappel::TrustRegionMaximizer, 782
- compute_bound_scaling_vec
 - mappel::TrustRegionMaximizer, 781
- compute_cauchy_point
 - mappel::TrustRegionMaximizer, 782
- compute_derivatives
 - mappel::Gauss1DModel::Stencil, 747
 - mappel::Gauss1DsModel::Stencil, 751
 - mappel::Gauss2DModel::Stencil, 766
 - mappel::Gauss2DsModel::Stencil, 760
 - mappel::Gauss2DsxyModel::Stencil, 755
- compute_estimate
 - mappel::CGaussHeuristicEstimator, 89
 - mappel::CGaussMLE, 98
 - mappel::Estimator, 106
 - mappel::HeuristicEstimator, 514
 - mappel::IterativeMaximizer, 536
 - mappel::NewtonDiagonalMaximizer, 662
 - mappel::NewtonMaximizer, 675
 - mappel::QuasiNewtonMaximizer, 727
 - mappel::SimulatedAnnealingMaximizer, 739, 740
 - mappel::ThreadedEstimator, 771
 - mappel::TrustRegionMaximizer, 782
- compute_estimate_debug
 - mappel::CGaussHeuristicEstimator, 90
 - mappel::CGaussMLE, 98
 - mappel::Estimator, 107
 - mappel::HeuristicEstimator, 514
 - mappel::IterativeMaximizer, 536
 - mappel::NewtonDiagonalMaximizer, 662
 - mappel::NewtonMaximizer, 675
 - mappel::QuasiNewtonMaximizer, 728
 - mappel::SimulatedAnnealingMaximizer, 740
 - mappel::ThreadedEstimator, 772
 - mappel::TrustRegionMaximizer, 782
- compute_initial_trust_radius
 - mappel::TrustRegionMaximizer, 783
- compute_max_sigma_ratio
 - mappel::Gauss2DsMAP, 375
 - mappel::Gauss2DsMLE, 407
 - mappel::Gauss2DsModel, 439
 - mappel::Gauss2DsxyModel, 495
- compute_profile_estimate
 - mappel::CGaussHeuristicEstimator, 90
 - mappel::CGaussMLE, 98
 - mappel::Estimator, 107
 - mappel::HeuristicEstimator, 514
 - mappel::IterativeMaximizer, 536
 - mappel::NewtonDiagonalMaximizer, 663
 - mappel::NewtonMaximizer, 676
 - mappel::QuasiNewtonMaximizer, 728
 - mappel::SimulatedAnnealingMaximizer, 740
 - mappel::ThreadedEstimator, 772
 - mappel::TrustRegionMaximizer, 783

- Continue
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [687](#)
- convergence_test
 - mappel::IterativeMaximizer, [537](#)
 - mappel::NewtonDiagonalMaximizer, [663](#)
 - mappel::NewtonMaximizer, [676](#)
 - mappel::QuasiNewtonMaximizer, [728](#)
 - mappel::TrustRegionMaximizer, [783](#)
- cooling_rate
 - mappel::SimulatedAnnealingMaximizer, [744](#)
- CoordIdxT
 - PoissonGaussianNoise2DObjective, [705](#)
- CoordStackT
 - PoissonGaussianNoise2DObjective, [705](#)
- CoordT
 - PoissonGaussianNoise2DObjective, [705](#)
- copy_Lsym_mat
 - mappel, [31](#)
- copy_Usym_mat
 - mappel, [31](#)
- copy_Usym_mat_stack
 - mappel, [31](#)
- cr_lower_bound
 - mappel::methods, [54](#)
- cr_lower_bound_stack
 - mappel, [31](#)
 - mappel::methods::openmp, [80](#)
- CubeT
 - mappel, [25](#)
- current_stencil
 - mappel::IterativeMaximizer::MaximizerData, [565](#)
- DXS2
 - mappel::Gauss1DsModel::Stencil, [752](#)
 - mappel::Gauss2DsModel::Stencil, [762](#)
 - mappel::Gauss2DsxyModel::Stencil, [756](#)
- DXSX
 - mappel::Gauss1DsModel::Stencil, [752](#)
 - mappel::Gauss2DsModel::Stencil, [762](#)
 - mappel::Gauss2DsxyModel::Stencil, [756](#)
- DXS
 - mappel::Gauss1DModel::Stencil, [748](#)
 - mappel::Gauss1DsModel::Stencil, [752](#)
 - mappel::Gauss2DModel::Stencil, [767](#)
 - mappel::Gauss2DsModel::Stencil, [762](#)
 - mappel::Gauss2DsxyModel::Stencil, [756](#)
- DYS2
 - mappel::Gauss2DsModel::Stencil, [763](#)
 - mappel::Gauss2DsxyModel::Stencil, [757](#)
- DYSX
 - mappel::Gauss2DsxyModel::Stencil, [757](#)
- DYSY
 - mappel::Gauss2DsModel::Stencil, [763](#)
 - mappel::Gauss2DsxyModel::Stencil, [757](#)
- DYS
 - mappel::Gauss2DsxyModel::Stencil, [757](#)
- debug_internal_sum_model_x
 - mappel::Gauss2DModel, [345](#)
 - mappel::Gauss2DsMAP, [375](#)
 - mappel::Gauss2DsMLE, [407](#)
 - mappel::Gauss2DsModel, [439](#)
- debug_internal_sum_model_y
 - mappel::Gauss2DModel, [346](#)
 - mappel::Gauss2DsMAP, [375](#)
 - mappel::Gauss2DsMLE, [407](#)
 - mappel::Gauss2DsModel, [439](#)
- default_alpha_sigma
 - mappel::Gauss1DModel, [188](#)
 - mappel::Gauss1DsMAP, [216](#)
 - mappel::Gauss1DsMLE, [244](#)
 - mappel::Gauss1DsModel, [272](#)
 - mappel::Gauss2DModel, [359](#)
 - mappel::Gauss2DsMAP, [391](#)
 - mappel::Gauss2DsMLE, [423](#)
 - mappel::Gauss2DsModel, [454](#)
 - mappel::Gauss2DsxyMAP, [482](#)
 - mappel::Gauss2DsxyModel, [506](#)
 - mappel::MAPEstimator, [557](#)
 - mappel::MCMCAdaptor1Ds, [599](#)
 - mappel::MCMCAdaptor1D, [581](#)
 - mappel::MCMCAdaptor2Ds, [636](#)
 - mappel::MCMCAdaptor2D, [617](#)
 - mappel::MLEstimator, [655](#)
 - mappel::PointEmitterModel, [701](#)
- default_beta_pos
 - mappel::Gauss1DModel, [188](#)
 - mappel::Gauss1DsMAP, [216](#)
 - mappel::Gauss1DsMLE, [245](#)
 - mappel::Gauss1DsModel, [272](#)
 - mappel::Gauss2DModel, [359](#)
 - mappel::Gauss2DsMAP, [391](#)
 - mappel::Gauss2DsMLE, [423](#)
 - mappel::Gauss2DsModel, [454](#)

- mappel::Gauss2DsxyMAP, [482](#)
- mappel::Gauss2DsxyModel, [506](#)
- mappel::MAPEstimator, [557](#)
- mappel::MCMCAdaptor1Ds, [599](#)
- mappel::MCMCAdaptor1D, [581](#)
- mappel::MCMCAdaptor2Ds, [636](#)
- mappel::MCMCAdaptor2D, [617](#)
- mappel::MLEstimator, [655](#)
- mappel::PointEmitterModel, [701](#)
- default_intensity_kappa
 - mappel::Gauss1DMAP, [133](#)
 - mappel::Gauss1DMLE, [161](#)
 - mappel::Gauss1DModel, [188](#)
 - mappel::Gauss1DsMAP, [217](#)
 - mappel::Gauss1DsMLE, [245](#)
 - mappel::Gauss1DsModel, [272](#)
 - mappel::Gauss2DMAP, [301](#)
 - mappel::Gauss2DMLE, [331](#)
 - mappel::Gauss2DModel, [359](#)
 - mappel::Gauss2DsMAP, [391](#)
 - mappel::Gauss2DsMLE, [423](#)
 - mappel::Gauss2DsModel, [454](#)
 - mappel::Gauss2DsxyMAP, [482](#)
 - mappel::Gauss2DsxyModel, [507](#)
 - mappel::MAPEstimator, [557](#)
 - mappel::MCMCAdaptor1Ds, [599](#)
 - mappel::MCMCAdaptor1D, [582](#)
 - mappel::MCMCAdaptor2Ds, [636](#)
 - mappel::MCMCAdaptor2D, [618](#)
 - mappel::MLEstimator, [655](#)
 - mappel::PointEmitterModel, [701](#)
- default_max_l
 - mappel::Gauss1DMAP, [133](#)
 - mappel::Gauss1DMLE, [161](#)
 - mappel::Gauss1DModel, [188](#)
 - mappel::Gauss1DsMAP, [217](#)
 - mappel::Gauss1DsMLE, [245](#)
 - mappel::Gauss1DsModel, [272](#)
 - mappel::Gauss2DMAP, [301](#)
 - mappel::Gauss2DMLE, [331](#)
 - mappel::Gauss2DModel, [360](#)
 - mappel::Gauss2DsMAP, [391](#)
 - mappel::Gauss2DsMLE, [423](#)
 - mappel::Gauss2DsModel, [454](#)
 - mappel::Gauss2DsxyMAP, [482](#)
 - mappel::Gauss2DsxyModel, [507](#)
 - mappel::MAPEstimator, [557](#)
 - mappel::MCMCAdaptor1Ds, [600](#)
 - mappel::MCMCAdaptor1D, [582](#)
 - mappel::MCMCAdaptor2Ds, [636](#)
 - mappel::MCMCAdaptor2D, [618](#)
 - mappel::MLEstimator, [655](#)
 - mappel::PointEmitterModel, [702](#)
- default_pixel_mean_bg
 - mappel::Gauss1DMAP, [134](#)
 - mappel::Gauss1DMLE, [162](#)
 - mappel::Gauss1DModel, [189](#)
 - mappel::Gauss1DsMAP, [217](#)
 - mappel::Gauss1DsMLE, [245](#)
 - mappel::Gauss1DsModel, [272](#)
 - mappel::Gauss2DMAP, [301](#)
 - mappel::Gauss2DMLE, [331](#)
 - mappel::Gauss2DModel, [360](#)
 - mappel::Gauss2DsMAP, [392](#)
 - mappel::Gauss2DsMLE, [424](#)
 - mappel::Gauss2DsModel, [455](#)
 - mappel::Gauss2DsxyMAP, [483](#)
 - mappel::Gauss2DsxyModel, [507](#)
 - mappel::MAPEstimator, [558](#)
 - mappel::MCMCAdaptor1Ds, [600](#)
 - mappel::MCMCAdaptor1D, [582](#)
 - mappel::MCMCAdaptor2Ds, [636](#)
 - mappel::MCMCAdaptor2D, [618](#)
 - mappel::MLEstimator, [656](#)
 - mappel::PointEmitterModel, [702](#)
- default_sigma_pos
 - mappel::Gauss1DMAP, [134](#)
 - mappel::Gauss1DMLE, [162](#)
 - mappel::Gauss1DModel, [189](#)
 - mappel::Gauss1DsMAP, [217](#)
 - mappel::Gauss1DsMLE, [245](#)
 - mappel::Gauss1DsModel, [273](#)
 - mappel::Gauss2DMAP, [301](#)
 - mappel::Gauss2DMLE, [331](#)
 - mappel::Gauss2DModel, [360](#)
 - mappel::Gauss2DsMAP, [392](#)

- mappel::Gauss2DsMLE, [424](#)
- mappel::Gauss2DsModel, [455](#)
- mappel::Gauss2DsxyMAP, [483](#)
- mappel::Gauss2DsxyModel, [507](#)
- mappel::MAPEstimator, [558](#)
- mappel::MCMCAdaptor1Ds, [600](#)
- mappel::MCMCAdaptor1D, [582](#)
- mappel::MCMCAdaptor2Ds, [637](#)
- mappel::MCMCAdaptor2D, [618](#)
- mappel::MLEstimator, [656](#)
- mappel::PointEmitterModel, [702](#)
- DefaultPriorType
 - mappel::Gauss1DModel, [134](#)
 - mappel::Gauss1DModel, [162](#)
 - mappel::Gauss1DModel, [189](#)
 - mappel::Gauss1DsMAP, [217](#)
 - mappel::Gauss1DsMLE, [246](#)
 - mappel::Gauss1DsModel, [273](#)
 - mappel::Gauss2DModel, [301](#)
 - mappel::Gauss2DModel, [331](#)
 - mappel::Gauss2DModel, [360](#)
 - mappel::Gauss2DsMAP, [392](#)
 - mappel::Gauss2DsMLE, [424](#)
 - mappel::Gauss2DsModel, [455](#)
 - mappel::Gauss2DsxyMAP, [483](#)
- DefaultSeperableInitEstimator
 - mappel::Gauss1DModel, [134](#)
 - mappel::Gauss1DModel, [162](#)
 - mappel::Gauss1DModel, [189](#)
 - mappel::Gauss1DsMAP, [218](#)
 - mappel::Gauss1DsMLE, [246](#)
 - mappel::Gauss1DsModel, [273](#)
 - mappel::Gauss2DModel, [302](#)
 - mappel::Gauss2DModel, [332](#)
 - mappel::Gauss2DModel, [360](#)
 - mappel::Gauss2DsMAP, [392](#)
 - mappel::Gauss2DsMLE, [424](#)
 - mappel::Gauss2DsModel, [455](#)
 - mappel::Gauss2DsxyMAP, [483](#)
 - mappel::Gauss2DsxyModel, [507](#)
 - mappel::MAPEstimator, [558](#)
 - mappel::MCMCAdaptor1Ds, [600](#)
 - mappel::MCMCAdaptor1D, [582](#)
 - mappel::MCMCAdaptor2Ds, [637](#)
 - mappel::MCMCAdaptor2D, [618](#)
 - mappel::MLEstimator, [656](#)
 - mappel::PointEmitterModel, [702](#)
- delta
 - mappel::IterativeMaximizer, [541](#)
 - mappel::NewtonDiagonalMaximizer, [668](#)
 - mappel::NewtonMaximizer, [681](#)
 - mappel::QuasiNewtonMaximizer, [733](#)
 - mappel::TrustRegionMaximizer, [789](#)
- delta_decrease
 - mappel::TrustRegionMaximizer, [789](#)
- delta_decrease_min
 - mappel::TrustRegionMaximizer, [789](#)
- delta_increase
 - mappel::TrustRegionMaximizer, [789](#)
- delta_init_max
 - mappel::TrustRegionMaximizer, [789](#)
- delta_init_min
 - mappel::TrustRegionMaximizer, [789](#)
- derivatives_computed
 - mappel::Gauss1DModel::Stencil, [748](#)
 - mappel::Gauss1DsModel::Stencil, [752](#)
 - mappel::Gauss2DModel::Stencil, [767](#)
 - mappel::Gauss2DsModel::Stencil, [761](#)
 - mappel::Gauss2DsxyModel::Stencil, [756](#)
- display.cpp, [793](#)
- display.h, [794](#)
- DoNotTry
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [687](#)
- DX
 - mappel::Gauss1DModel::Stencil, [748](#)
 - mappel::Gauss1DsModel::Stencil, [752](#)
 - mappel::Gauss2DModel::Stencil, [767](#)
 - mappel::Gauss2DsModel::Stencil, [762](#)
 - mappel::Gauss2DsxyModel::Stencil, [756](#)
- dx
 - mappel::Gauss1DModel::Stencil, [748](#)
 - mappel::Gauss1DsModel::Stencil, [752](#)
 - mappel::Gauss2DModel::Stencil, [767](#)
 - mappel::Gauss2DsModel::Stencil, [761](#)
 - mappel::Gauss2DsxyModel::Stencil, [756](#)
- DY
 - mappel::Gauss2DModel::Stencil, [768](#)
 - mappel::Gauss2DsModel::Stencil, [762](#)
 - mappel::Gauss2DsxyModel::Stencil, [757](#)
- dy
 - mappel::Gauss2DModel::Stencil, [767](#)
 - mappel::Gauss2DsModel::Stencil, [762](#)
 - mappel::Gauss2DsxyModel::Stencil, [757](#)
- enable_all_cpus
 - mappel, [31](#)
- EnableIfSubclassT
 - mappel, [25](#)
- epsilon
 - mappel::IterativeMaximizer, [541](#)
 - mappel::NewtonDiagonalMaximizer, [668](#)
 - mappel::NewtonMaximizer, [681](#)
 - mappel::QuasiNewtonMaximizer, [733](#)
 - mappel::TrustRegionMaximizer, [789](#)
- Error
 - mappel::CGaussHeuristicEstimator, [89](#)
 - mappel::CGaussMLE, [97](#)

- mappel::Estimator, 105
- mappel::HeuristicEstimator, 513
- mappel::IterativeMaximizer, 535
- mappel::NewtonDiagonalMaximizer, 661
- mappel::NewtonMaximizer, 674
- mappel::QuasiNewtonMaximizer, 726
- mappel::SimulatedAnnealingMaximizer, 739
- mappel::ThreadedEstimator, 771
- mappel::TrustRegionMaximizer, 780
- error_bounds_expected
 - mappel::methods, 54
- error_bounds_expected_stack
 - mappel::methods::openmp, 80
- error_bounds_observed
 - mappel::methods, 54
- error_bounds_observed_stack
 - mappel::methods::openmp, 81
- error_bounds_posterior_credible
 - mappel::methods, 54
- error_bounds_posterior_credible_stack
 - mappel::methods::openmp, 81
- estimate_background
 - mappel, 31
- estimate_gaussian_2Dmax
 - mappel, 32
- estimate_gaussian_3Dmax
 - mappel, 32
- estimate_intensity
 - mappel, 32
- estimate_max
 - mappel::CGaussHeuristicEstimator, 90, 91
 - mappel::CGaussMLE, 99
 - mappel::Estimator, 107, 108
 - mappel::HeuristicEstimator, 514, 515
 - mappel::IterativeMaximizer, 537, 538
 - mappel::NewtonDiagonalMaximizer, 663, 664
 - mappel::NewtonMaximizer, 676, 677
 - mappel::QuasiNewtonMaximizer, 728, 729
 - mappel::SimulatedAnnealingMaximizer, 740, 741
 - mappel::ThreadedEstimator, 772, 773
 - mappel::TrustRegionMaximizer, 783, 784
 - mappel::methods, 54, 55
- estimate_max_debug
 - mappel::CGaussHeuristicEstimator, 91
 - mappel::CGaussMLE, 100
 - mappel::Estimator, 108
 - mappel::HeuristicEstimator, 515
 - mappel::IterativeMaximizer, 538
 - mappel::NewtonDiagonalMaximizer, 664
 - mappel::NewtonMaximizer, 677
 - mappel::QuasiNewtonMaximizer, 729
 - mappel::SimulatedAnnealingMaximizer, 741
 - mappel::ThreadedEstimator, 773
 - mappel::TrustRegionMaximizer, 784
- estimate_max_stack
 - mappel::CGaussHeuristicEstimator, 91, 92
 - mappel::CGaussMLE, 100
 - mappel::Estimator, 108, 109
 - mappel::HeuristicEstimator, 516
 - mappel::IterativeMaximizer, 538
 - mappel::NewtonDiagonalMaximizer, 664, 665
 - mappel::NewtonMaximizer, 677, 678
 - mappel::QuasiNewtonMaximizer, 730
 - mappel::SimulatedAnnealingMaximizer, 741, 742
 - mappel::ThreadedEstimator, 773, 774
 - mappel::TrustRegionMaximizer, 785
 - mappel::methods::openmp, 81
- estimate_mcmc_posterior
 - mappel::methods, 55
- estimate_mcmc_posterior_stack
 - mappel::methods::openmp, 82
- estimate_mcmc_sample
 - mappel::methods, 56
- estimate_mcmc_sample_debug
 - mappel::methods::debug, 62
- estimate_mcmc_sample_stack
 - mappel::methods::openmp, 82
- estimate_profile_likelihood
 - mappel::methods::openmp, 82, 83
- estimate_profile_stack
 - mappel::CGaussHeuristicEstimator, 92
 - mappel::CGaussMLE, 100
 - mappel::Estimator, 109
 - mappel::HeuristicEstimator, 516
 - mappel::IterativeMaximizer, 539
 - mappel::NewtonDiagonalMaximizer, 665
 - mappel::NewtonMaximizer, 678
 - mappel::QuasiNewtonMaximizer, 730
 - mappel::SimulatedAnnealingMaximizer, 742
 - mappel::ThreadedEstimator, 774
 - mappel::TrustRegionMaximizer, 785
- estimate_sample_posterior
 - mappel::mcmc, 49
- Estimator
 - mappel::Estimator, 106
- estimator.h, 795
 - ClockT, 796
- estimator_impl.h, 796
- estimator_names
 - mappel::Gauss1DMAP, 134
 - mappel::Gauss1DMLE, 162
 - mappel::Gauss1DsMAP, 218
 - mappel::Gauss1DsMLE, 246
 - mappel::Gauss2DMAP, 302
 - mappel::Gauss2DMLE, 332
 - mappel::Gauss2DsMAP, 392
 - mappel::Gauss2DsMLE, 424

- mappel::Gauss2DsxyMAP, [483](#)
- mappel::PoissonNoise1DObjective, [712](#)
- mappel::PoissonNoise2DObjective, [720](#)
- PoissonGaussianNoise2DObjective, [706](#)
- estimator_statics.cpp, [796](#)
- eta_bg
 - mappel::Gauss1DMAP, [135](#)
 - mappel::Gauss1DMLE, [163](#)
 - mappel::Gauss1DModel, [189](#)
 - mappel::Gauss1DsMAP, [218](#)
 - mappel::Gauss1DsMLE, [246](#)
 - mappel::Gauss1DsModel, [273](#)
 - mappel::Gauss2DMAP, [302](#)
 - mappel::Gauss2DMLE, [332](#)
 - mappel::Gauss2DModel, [361](#)
 - mappel::Gauss2DsMAP, [392](#)
 - mappel::Gauss2DsMLE, [424](#)
 - mappel::Gauss2DsModel, [455](#)
 - mappel::Gauss2DsxyMAP, [484](#)
 - mappel::MCMCAdaptor1Ds, [600](#)
 - mappel::MCMCAdaptor1D, [583](#)
 - mappel::MCMCAdaptor2Ds, [637](#)
 - mappel::MCMCAdaptor2D, [619](#)
- eta_l
 - mappel::Gauss1DMAP, [135](#)
 - mappel::Gauss1DMLE, [163](#)
 - mappel::Gauss1DModel, [190](#)
 - mappel::Gauss1DsMAP, [218](#)
 - mappel::Gauss1DsMLE, [246](#)
 - mappel::Gauss1DsModel, [273](#)
 - mappel::Gauss2DMAP, [302](#)
 - mappel::Gauss2DMLE, [332](#)
 - mappel::Gauss2DModel, [361](#)
 - mappel::Gauss2DsMAP, [393](#)
 - mappel::Gauss2DsMLE, [425](#)
 - mappel::Gauss2DsModel, [455](#)
 - mappel::Gauss2DsxyMAP, [484](#)
 - mappel::MCMCAdaptor1Ds, [601](#)
 - mappel::MCMCAdaptor1D, [583](#)
 - mappel::MCMCAdaptor2Ds, [637](#)
 - mappel::MCMCAdaptor2D, [619](#)
- eta_sigma
 - mappel::Gauss1DsMAP, [218](#)
 - mappel::Gauss1DsMLE, [246](#)
 - mappel::Gauss1DsModel, [273](#)
 - mappel::Gauss2DsMAP, [393](#)
 - mappel::Gauss2DsMLE, [425](#)
 - mappel::Gauss2DsModel, [456](#)
 - mappel::MCMCAdaptor1Ds, [601](#)
 - mappel::MCMCAdaptor2Ds, [637](#)
- eta_x
 - mappel::Gauss1DMAP, [135](#)
 - mappel::Gauss1DMLE, [163](#)
 - mappel::Gauss1DModel, [190](#)
- mappel::Gauss1DsMAP, [218](#)
- mappel::Gauss1DsMLE, [247](#)
- mappel::Gauss1DsModel, [274](#)
- mappel::Gauss2DMAP, [302](#)
- mappel::Gauss2DMLE, [332](#)
- mappel::Gauss2DModel, [361](#)
- mappel::Gauss2DsMAP, [393](#)
- mappel::Gauss2DsMLE, [425](#)
- mappel::Gauss2DsModel, [456](#)
- mappel::Gauss2DsxyMAP, [484](#)
- mappel::MCMCAdaptor1Ds, [601](#)
- mappel::MCMCAdaptor1D, [583](#)
- mappel::MCMCAdaptor2Ds, [638](#)
- mappel::MCMCAdaptor2D, [619](#)
- eta_y
 - mappel::Gauss2DMAP, [302](#)
 - mappel::Gauss2DMLE, [332](#)
 - mappel::Gauss2DModel, [361](#)
 - mappel::Gauss2DsMAP, [393](#)
 - mappel::Gauss2DsMLE, [425](#)
 - mappel::Gauss2DsModel, [456](#)
 - mappel::MCMCAdaptor2Ds, [638](#)
 - mappel::MCMCAdaptor2D, [619](#)
- exit_counts
 - mappel::CGaussHeuristicEstimator, [94](#)
 - mappel::CGaussMLE, [102](#)
 - mappel::Estimator, [110](#)
 - mappel::HeuristicEstimator, [518](#)
 - mappel::IterativeMaximizer, [542](#)
 - mappel::NewtonDiagonalMaximizer, [668](#)
 - mappel::NewtonMaximizer, [681](#)
 - mappel::QuasiNewtonMaximizer, [733](#)
 - mappel::SimulatedAnnealingMaximizer, [744](#)
 - mappel::ThreadedEstimator, [776](#)
 - mappel::TrustRegionMaximizer, [789](#)
- ExitCode
 - mappel::CGaussHeuristicEstimator, [89](#)
 - mappel::CGaussMLE, [97](#)
 - mappel::Estimator, [105](#)
 - mappel::HeuristicEstimator, [513](#)
 - mappel::IterativeMaximizer, [535](#)
 - mappel::NewtonDiagonalMaximizer, [661](#)
 - mappel::NewtonMaximizer, [674](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::SimulatedAnnealingMaximizer, [739](#)
 - mappel::ThreadedEstimator, [771](#)
 - mappel::TrustRegionMaximizer, [780](#)
- expected_information
 - mappel::methods, [56](#), [57](#)
- expected_information_stack
 - mappel::methods::openmp, [83](#)
- fill_DX_stencil
 - mappel, [32](#)

- fill_DXS2_stencil
 - mappel, [32](#)
- fill_DXS_stencil
 - mappel, [32](#)
- fill_DXSX_stencil
 - mappel, [33](#)
- fill_G_stencil
 - mappel, [33](#)
- fill_X_stencil
 - mappel, [33](#)
- fill_d_stencil
 - mappel, [32](#)
- fill_gaussian_stencil
 - mappel, [33](#)
- fisher_information
 - PoissonGaussianNoise2DObjective.h, [842](#)
- fisher_information_stack
 - mappel, [33](#)
- fixed_idxs
 - mappel::IterativeMaximizer::MaximizerData, [565](#)
- free_parameters
 - mappel::IterativeMaximizer::MaximizerData, [565](#)
- FunctionChange
 - mappel::CGaussHeuristicEstimator, [89](#)
 - mappel::CGaussMLE, [97](#)
 - mappel::Estimator, [105](#)
 - mappel::HeuristicEstimator, [513](#)
 - mappel::IterativeMaximizer, [535](#)
 - mappel::NewtonDiagonalMaximizer, [661](#)
 - mappel::NewtonMaximizer, [674](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::SimulatedAnnealingMaximizer, [739](#)
 - mappel::ThreadedEstimator, [771](#)
 - mappel::TrustRegionMaximizer, [780](#)
- gamma_prior_grad
 - mappel, [33](#)
- gamma_prior_grad2
 - mappel, [33](#)
- Gauss1DMAP.cpp, [797](#)
- Gauss1DMAP.h, [797](#)
- Gauss1DMAP
 - mappel::Gauss1DMAP, [118](#)
- Gauss1DMLE.cpp, [798](#)
- Gauss1DMLE.h, [799](#)
- Gauss1DMLE
 - mappel::Gauss1DMLE, [145](#), [146](#)
- Gauss1DModel
 - mappel::Gauss1DModel, [173](#)
- Gauss1DModel.cpp, [799](#)
- Gauss1DModel.h, [800](#)
- Gauss1DSumModelT
 - mappel::Gauss2DMAP, [282](#)
 - mappel::Gauss2DMLE, [312](#)
- mappel::Gauss2DModel, [341](#)
- mappel::Gauss2DsMAP, [370](#)
- mappel::Gauss2DsMLE, [402](#)
- mappel::Gauss2DsModel, [435](#)
- Gauss1DsMAP.cpp, [801](#)
- Gauss1DsMAP.h, [801](#)
- Gauss1DsMAP
 - mappel::Gauss1DsMAP, [200](#), [201](#)
- Gauss1DsMLE.cpp, [802](#)
- Gauss1DsMLE.h, [802](#)
- Gauss1DsMLE
 - mappel::Gauss1DsMLE, [229](#)
- Gauss1DsModel
 - mappel::Gauss1DsModel, [256](#)
- Gauss1DsModel.cpp, [803](#)
- Gauss1DsModel.h, [804](#)
- Gauss2DMAP.cpp, [804](#)
- Gauss2DMAP.h, [805](#)
- Gauss2DMAP
 - mappel::Gauss2DMAP, [284](#), [285](#)
- Gauss2DMLE.cpp, [806](#)
- Gauss2DMLE.h, [806](#)
- Gauss2DMLE
 - mappel::Gauss2DMLE, [314](#)
- Gauss2DModel
 - mappel::Gauss2DModel, [343](#), [344](#)
- Gauss2DModel.cpp, [807](#)
- Gauss2DModel.h, [808](#)
- Gauss2DsMAP.cpp, [809](#)
- Gauss2DsMAP.h, [809](#)
- Gauss2DsMAP
 - mappel::Gauss2DsMAP, [373](#)
- Gauss2DsMLE.cpp, [810](#)
- Gauss2DsMLE.h, [810](#)
- Gauss2DsMLE
 - mappel::Gauss2DsMLE, [405](#)
- Gauss2DsModel
 - mappel::Gauss2DsModel, [437](#)
- Gauss2DsModel.cpp, [811](#)
- Gauss2DsModel.h, [812](#)
- Gauss2DsxyMAP.h, [813](#)
- Gauss2DsxyMAP
 - mappel::Gauss2DsxyMAP, [467](#)
- Gauss2DsxyModel
 - mappel::Gauss2DsxyModel, [494](#)
- Gauss2DsxyModel.h, [813](#)
- gauss_norm
 - mappel, [34](#)
- gaussian_3D_convolution
 - mappel, [34](#)
- gaussian_convolution
 - mappel, [34](#)
- generate_poisson
 - mappel, [34](#)

- generate_poisson_large
 - mappel, [34](#)
- generate_poisson_small
 - mappel, [34](#)
- get_backtrack_idx
 - mappel::IterativeMaximizer::MaximizerData, [562](#)
- get_debug_stats
 - mappel::CGaussHeuristicEstimator, [92](#)
 - mappel::CGaussMLE, [100](#)
 - mappel::Estimator, [109](#)
 - mappel::HeuristicEstimator, [516](#)
 - mappel::IterativeMaximizer, [539](#)
 - mappel::NewtonDiagonalMaximizer, [665](#)
 - mappel::NewtonMaximizer, [678](#)
 - mappel::QuasiNewtonMaximizer, [730](#)
 - mappel::SimulatedAnnealingMaximizer, [742](#)
 - mappel::ThreadedEstimator, [774](#)
 - mappel::TrustRegionMaximizer, [785](#)
- get_hyperparam_index
 - mappel::Gauss1DMAP, [120](#)
 - mappel::Gauss1DMLE, [147](#)
 - mappel::Gauss1DModel, [175](#)
 - mappel::Gauss1DsMAP, [202](#)
 - mappel::Gauss1DsMLE, [231](#)
 - mappel::Gauss1DsModel, [258](#)
 - mappel::Gauss2DMAP, [287](#)
 - mappel::Gauss2DMLE, [317](#)
 - mappel::Gauss2DModel, [346](#)
 - mappel::Gauss2DsMAP, [376](#)
 - mappel::Gauss2DsMLE, [408](#)
 - mappel::Gauss2DsModel, [439](#)
 - mappel::Gauss2DsxyMAP, [469](#)
 - mappel::Gauss2DsxyModel, [496](#)
 - mappel::MAPEstimator, [549](#)
 - mappel::MCMCAdaptor1Ds, [590](#)
 - mappel::MCMCAdaptor1D, [572](#)
 - mappel::MCMCAdaptor2Ds, [627](#)
 - mappel::MCMCAdaptor2D, [608](#)
 - mappel::MLEstimator, [647](#)
 - mappel::PointEmitterModel, [694](#)
- get_hyperparam_names
 - mappel::Gauss1DMAP, [120](#)
 - mappel::Gauss1DMLE, [148](#)
 - mappel::Gauss1DModel, [175](#)
 - mappel::Gauss1DsMAP, [203](#)
 - mappel::Gauss1DsMLE, [231](#)
 - mappel::Gauss1DsModel, [258](#)
 - mappel::Gauss2DMAP, [287](#)
 - mappel::Gauss2DMLE, [317](#)
 - mappel::Gauss2DModel, [346](#)
 - mappel::Gauss2DsMAP, [376](#)
 - mappel::Gauss2DsMLE, [408](#)
 - mappel::Gauss2DsModel, [440](#)
 - mappel::Gauss2DsxyMAP, [469](#)
 - mappel::Gauss2DsxyModel, [496](#)
 - mappel::MAPEstimator, [550](#)
 - mappel::MCMCAdaptor1Ds, [591](#)
 - mappel::MCMCAdaptor1D, [573](#)
 - mappel::MCMCAdaptor2Ds, [627](#)
 - mappel::MCMCAdaptor2D, [609](#)
 - mappel::MLEstimator, [648](#)
 - mappel::PointEmitterModel, [694](#)
- mappel::Gauss2DsxyModel, [496](#)
- mappel::MAPEstimator, [550](#)
- mappel::MCMCAdaptor1Ds, [591](#)
- mappel::MCMCAdaptor1D, [572](#)
- mappel::MCMCAdaptor2Ds, [627](#)
- mappel::MCMCAdaptor2D, [609](#)
- mappel::MLEstimator, [648](#)
- mappel::PointEmitterModel, [694](#)

- get_hyperparam_value
 - mappel::Gauss1DMAP, [120](#)
 - mappel::Gauss1DMLE, [148](#)
 - mappel::Gauss1DModel, [175](#)
 - mappel::Gauss1DsMAP, [203](#)
 - mappel::Gauss1DsMLE, [231](#)
 - mappel::Gauss1DsModel, [258](#)
 - mappel::Gauss2DMAP, [287](#)
 - mappel::Gauss2DMLE, [317](#)
 - mappel::Gauss2DModel, [346](#)
 - mappel::Gauss2DsMAP, [376](#)
 - mappel::Gauss2DsMLE, [408](#)
 - mappel::Gauss2DsModel, [440](#)
 - mappel::Gauss2DsxyMAP, [469](#)
 - mappel::Gauss2DsxyModel, [496](#)
 - mappel::MAPEstimator, [550](#)
 - mappel::MCMCAdaptor1Ds, [591](#)
 - mappel::MCMCAdaptor1D, [573](#)
 - mappel::MCMCAdaptor2Ds, [627](#)
 - mappel::MCMCAdaptor2D, [609](#)
 - mappel::MLEstimator, [648](#)
 - mappel::PointEmitterModel, [694](#)
- get_hyperparams
 - mappel::Gauss1DMAP, [120](#)
 - mappel::Gauss1DMLE, [148](#)
 - mappel::Gauss1DModel, [175](#)
 - mappel::Gauss1DsMAP, [203](#)
 - mappel::Gauss1DsMLE, [231](#)
 - mappel::Gauss1DsModel, [259](#)
 - mappel::Gauss2DMAP, [287](#)
 - mappel::Gauss2DMLE, [317](#)
 - mappel::Gauss2DModel, [346](#)
 - mappel::Gauss2DsMAP, [376](#)
 - mappel::Gauss2DsMLE, [408](#)
 - mappel::Gauss2DsModel, [440](#)
 - mappel::Gauss2DsxyMAP, [469](#)
 - mappel::Gauss2DsxyModel, [496](#)
 - mappel::MAPEstimator, [550](#)
 - mappel::MCMCAdaptor1Ds, [591](#)
 - mappel::MCMCAdaptor1D, [573](#)
 - mappel::MCMCAdaptor2Ds, [627](#)
 - mappel::MCMCAdaptor2D, [609](#)
 - mappel::MLEstimator, [648](#)
 - mappel::PointEmitterModel, [694](#)
- get_image_from_stack
 - mappel::Gauss1DMAP, [120](#)

- mappel::Gauss1DMLE, [148](#)
- mappel::Gauss1DModel, [176](#)
- mappel::Gauss1DsMAP, [203](#)
- mappel::Gauss1DsMLE, [231](#)
- mappel::Gauss1DsModel, [259](#)
- mappel::Gauss2DMAP, [287](#)
- mappel::Gauss2DMLE, [317](#)
- mappel::Gauss2DModel, [346](#)
- mappel::Gauss2DsMAP, [376](#)
- mappel::Gauss2DsMLE, [408](#)
- mappel::Gauss2DsModel, [440](#)
- mappel::Gauss2DsxyMAP, [470](#)
- mappel::Gauss2DsxyModel, [496](#)
- mappel::ImageFormat1DBase, [523](#)
- mappel::ImageFormat2DBase, [529](#)
- mappel::PoissonNoise1DObjective, [710](#)
- mappel::PoissonNoise2DObjective, [718](#)
- get_lbound
 - mappel::Gauss1DMAP, [120](#)
 - mappel::Gauss1DMLE, [148](#)
 - mappel::Gauss1DModel, [176](#)
 - mappel::Gauss1DsMAP, [203](#)
 - mappel::Gauss1DsMLE, [231](#)
 - mappel::Gauss1DsModel, [259](#)
 - mappel::Gauss2DMAP, [287](#)
 - mappel::Gauss2DMLE, [317](#)
 - mappel::Gauss2DModel, [346](#)
 - mappel::Gauss2DsMAP, [376](#)
 - mappel::Gauss2DsMLE, [408](#)
 - mappel::Gauss2DsModel, [440](#)
 - mappel::Gauss2DsxyMAP, [470](#)
 - mappel::Gauss2DsxyModel, [496](#)
 - mappel::MAPEstimator, [550](#)
 - mappel::MCMCAdaptor1Ds, [591](#)
 - mappel::MCMCAdaptor1D, [573](#)
 - mappel::MCMCAdaptor2Ds, [628](#)
 - mappel::MCMCAdaptor2D, [609](#)
 - mappel::MLEstimator, [648](#)
 - mappel::PointEmitterModel, [694](#)
- get_max_sigma
 - mappel::Gauss1DsMAP, [203](#)
 - mappel::Gauss1DsMLE, [232](#)
 - mappel::Gauss1DsModel, [259](#)
 - mappel::Gauss2DsMAP, [376](#), [377](#)
 - mappel::Gauss2DsMLE, [408](#), [409](#)
 - mappel::Gauss2DsModel, [440](#)
 - mappel::Gauss2DsxyModel, [496](#), [497](#)
- get_max_sigma_ratio
 - mappel::Gauss2DsMAP, [377](#)
 - mappel::Gauss2DsMLE, [409](#)
 - mappel::Gauss2DsModel, [441](#)
 - mappel::Gauss2DsxyModel, [497](#)
- get_mcmc_num_phases
 - mappel::Gauss1DMAP, [121](#)
- mappel::Gauss1DMLE, [148](#)
- mappel::Gauss1DModel, [176](#)
- mappel::Gauss1DsMAP, [203](#)
- mappel::Gauss1DsMLE, [232](#)
- mappel::Gauss1DsModel, [259](#)
- mappel::Gauss2DMAP, [287](#)
- mappel::Gauss2DMLE, [317](#)
- mappel::Gauss2DModel, [347](#)
- mappel::Gauss2DsMAP, [377](#)
- mappel::Gauss2DsMLE, [409](#)
- mappel::Gauss2DsModel, [441](#)
- mappel::Gauss2DsxyMAP, [470](#)
- mappel::MCMCAdaptor1Ds, [591](#)
- mappel::MCMCAdaptor1D, [573](#)
- mappel::MCMCAdaptor2Ds, [628](#)
- mappel::MCMCAdaptor2D, [609](#)
- mappel::MCMCAdaptorBase, [642](#)
- get_mcmc_sigma_scale
 - mappel::Gauss1DMAP, [121](#)
 - mappel::Gauss1DMLE, [148](#)
 - mappel::Gauss1DModel, [176](#)
 - mappel::Gauss1DsMAP, [204](#)
 - mappel::Gauss1DsMLE, [232](#)
 - mappel::Gauss1DsModel, [259](#)
 - mappel::Gauss2DMAP, [288](#)
 - mappel::Gauss2DMLE, [318](#)
 - mappel::Gauss2DModel, [347](#)
 - mappel::Gauss2DsMAP, [377](#)
 - mappel::Gauss2DsMLE, [409](#)
 - mappel::Gauss2DsModel, [441](#)
 - mappel::Gauss2DsxyMAP, [470](#)
 - mappel::MCMCAdaptor1Ds, [591](#)
 - mappel::MCMCAdaptor1D, [573](#)
 - mappel::MCMCAdaptor2Ds, [628](#)
 - mappel::MCMCAdaptor2D, [609](#)
 - mappel::MCMCAdaptorBase, [642](#)
- get_min_sigma
 - mappel::Gauss1DsMAP, [204](#)
 - mappel::Gauss1DsMLE, [232](#)
 - mappel::Gauss1DsModel, [259](#)
 - mappel::Gauss2DsMAP, [377](#)
 - mappel::Gauss2DsMLE, [409](#)
 - mappel::Gauss2DsModel, [441](#)
 - mappel::Gauss2DsxyModel, [497](#)
- get_model
 - mappel::CGaussHeuristicEstimator, [92](#)
 - mappel::CGaussMLE, [101](#)
 - mappel::Estimator, [109](#)
 - mappel::HeuristicEstimator, [516](#)
 - mappel::IterativeMaximizer, [539](#)
 - mappel::NewtonDiagonalMaximizer, [665](#)
 - mappel::NewtonMaximizer, [678](#)
 - mappel::QuasiNewtonMaximizer, [730](#)
 - mappel::SimulatedAnnealingMaximizer, [742](#)

mappel::ThreadedEstimator, 774
 mappel::TrustRegionMaximizer, 785

get_num_hyperparams

mappel::Gauss1DMAP, 121
 mappel::Gauss1DMLE, 149
 mappel::Gauss1DModel, 176
 mappel::Gauss1DsMAP, 204
 mappel::Gauss1DsMLE, 232
 mappel::Gauss1DsModel, 260
 mappel::Gauss2DMAP, 288
 mappel::Gauss2DMLE, 318
 mappel::Gauss2DModel, 347
 mappel::Gauss2DsMAP, 377
 mappel::Gauss2DsMLE, 409
 mappel::Gauss2DsModel, 441
 mappel::Gauss2DsxyMAP, 470
 mappel::Gauss2DsxyModel, 497
 mappel::MAPEstimator, 550
 mappel::MCMCAdaptor1Ds, 591
 mappel::MCMCAdaptor1D, 573
 mappel::MCMCAdaptor2Ds, 628
 mappel::MCMCAdaptor2D, 609
 mappel::MLEstimator, 648
 mappel::PointEmitterModel, 695

get_num_params

mappel::Gauss1DMAP, 121
 mappel::Gauss1DMLE, 149
 mappel::Gauss1DModel, 176
 mappel::Gauss1DsMAP, 204
 mappel::Gauss1DsMLE, 232
 mappel::Gauss1DsModel, 260
 mappel::Gauss2DMAP, 288
 mappel::Gauss2DMLE, 318
 mappel::Gauss2DModel, 347
 mappel::Gauss2DsMAP, 378
 mappel::Gauss2DsMLE, 410
 mappel::Gauss2DsModel, 441
 mappel::Gauss2DsxyMAP, 470
 mappel::Gauss2DsxyModel, 497
 mappel::MAPEstimator, 550
 mappel::MCMCAdaptor1Ds, 592
 mappel::MCMCAdaptor1D, 573
 mappel::MCMCAdaptor2Ds, 628
 mappel::MCMCAdaptor2D, 610
 mappel::MLEstimator, 648
 mappel::PointEmitterModel, 695

get_num_pixels

mappel::Gauss1DMAP, 121
 mappel::Gauss1DMLE, 149
 mappel::Gauss1DModel, 176
 mappel::Gauss1DsMAP, 204
 mappel::Gauss1DsMLE, 232
 mappel::Gauss1DsModel, 260
 mappel::Gauss2DMAP, 288

mappel::Gauss2DMLE, 318
 mappel::Gauss2DModel, 347
 mappel::Gauss2DsMAP, 378
 mappel::Gauss2DsMLE, 410
 mappel::Gauss2DsModel, 441
 mappel::Gauss2DsxyMAP, 470
 mappel::Gauss2DsxyModel, 497
 mappel::ImageFormat1DBase, 523
 mappel::ImageFormat2DBase, 530
 mappel::PoissonNoise1DObjective, 711
 mappel::PoissonNoise2DObjective, 718

get_param_names

mappel::Gauss1DMAP, 121
 mappel::Gauss1DMLE, 149
 mappel::Gauss1DModel, 177
 mappel::Gauss1DsMAP, 204
 mappel::Gauss1DsMLE, 233
 mappel::Gauss1DsModel, 260
 mappel::Gauss2DMAP, 288
 mappel::Gauss2DMLE, 318
 mappel::Gauss2DModel, 347
 mappel::Gauss2DsMAP, 378
 mappel::Gauss2DsMLE, 410
 mappel::Gauss2DsModel, 442
 mappel::Gauss2DsxyMAP, 471
 mappel::Gauss2DsxyModel, 497
 mappel::MAPEstimator, 550
 mappel::MCMCAdaptor1Ds, 592
 mappel::MCMCAdaptor1D, 574
 mappel::MCMCAdaptor2Ds, 628
 mappel::MCMCAdaptor2D, 610
 mappel::MLEstimator, 648
 mappel::PointEmitterModel, 695

get_prior

mappel::Gauss1DMAP, 121, 122
 mappel::Gauss1DMLE, 149
 mappel::Gauss1DModel, 177
 mappel::Gauss1DsMAP, 204, 205
 mappel::Gauss1DsMLE, 233
 mappel::Gauss1DsModel, 260
 mappel::Gauss2DMAP, 288
 mappel::Gauss2DMLE, 318
 mappel::Gauss2DModel, 347, 348
 mappel::Gauss2DsMAP, 378
 mappel::Gauss2DsMLE, 410
 mappel::Gauss2DsModel, 442
 mappel::Gauss2DsxyMAP, 471
 mappel::Gauss2DsxyModel, 498
 mappel::MAPEstimator, 551
 mappel::MCMCAdaptor1Ds, 592
 mappel::MCMCAdaptor1D, 574
 mappel::MCMCAdaptor2Ds, 628, 629
 mappel::MCMCAdaptor2D, 610
 mappel::MLEstimator, 649

- mappel::PointEmitterModel, 695
- get_psf_sigma
 - mappel::Gauss1DMAP, 122
 - mappel::Gauss1DMLE, 149, 150
 - mappel::Gauss1DModel, 177
 - mappel::Gauss2DMAP, 289
 - mappel::Gauss2DMLE, 319
 - mappel::Gauss2DModel, 348
 - mappel::Gauss2DsxyMAP, 471
- get_rng_generator
 - mappel::Gauss1DMAP, 122
 - mappel::Gauss1DMLE, 150
 - mappel::Gauss1DModel, 177
 - mappel::Gauss1DsMAP, 205
 - mappel::Gauss1DsMLE, 233
 - mappel::Gauss1DsModel, 260
 - mappel::Gauss2DMAP, 289
 - mappel::Gauss2DMLE, 319
 - mappel::Gauss2DModel, 348
 - mappel::Gauss2DsMAP, 378
 - mappel::Gauss2DsMLE, 410
 - mappel::Gauss2DsModel, 442
 - mappel::Gauss2DsxyMAP, 471
 - mappel::Gauss2DsxyModel, 498
 - mappel::MAPEstimator, 551
 - mappel::MCMCAdaptor1Ds, 592
 - mappel::MCMCAdaptor1D, 574
 - mappel::MCMCAdaptor2Ds, 629
 - mappel::MCMCAdaptor2D, 610
 - mappel::MLEstimator, 649
 - mappel::PointEmitterModel, 695
- get_rng_manager
 - mappel::Gauss1DMAP, 122
 - mappel::Gauss1DMLE, 150
 - mappel::Gauss1DModel, 177
 - mappel::Gauss1DsMAP, 205
 - mappel::Gauss1DsMLE, 233
 - mappel::Gauss1DsModel, 261
 - mappel::Gauss2DMAP, 289
 - mappel::Gauss2DMLE, 319
 - mappel::Gauss2DModel, 348
 - mappel::Gauss2DsMAP, 378
 - mappel::Gauss2DsMLE, 410
 - mappel::Gauss2DsModel, 442
 - mappel::Gauss2DsxyMAP, 471
 - mappel::Gauss2DsxyModel, 498
 - mappel::MAPEstimator, 551
 - mappel::MCMCAdaptor1Ds, 592
 - mappel::MCMCAdaptor1D, 574
 - mappel::MCMCAdaptor2Ds, 629
 - mappel::MCMCAdaptor2D, 610
 - mappel::MLEstimator, 649
 - mappel::PointEmitterModel, 695
- get_size

- mappel::Gauss1DMAP, 122
- mappel::Gauss1DMLE, 150
- mappel::Gauss1DModel, 178
- mappel::Gauss1DsMAP, 205
- mappel::Gauss1DsMLE, 233
- mappel::Gauss1DsModel, 261
- mappel::Gauss2DMAP, 289
- mappel::Gauss2DMLE, 319
- mappel::Gauss2DModel, 348
- mappel::Gauss2DsMAP, 379
- mappel::Gauss2DsMLE, 411
- mappel::Gauss2DsModel, 442
- mappel::Gauss2DsxyMAP, 472
- mappel::Gauss2DsxyModel, 498
- mappel::ImageFormat1DBase, 523
- mappel::ImageFormat2DBase, 530
- mappel::PoissonNoise1DObjective, 711
- mappel::PoissonNoise2DObjective, 718

- get_size_image_stack
 - mappel::Gauss1DMAP, 123
 - mappel::Gauss1DMLE, 150
 - mappel::Gauss1DModel, 178
 - mappel::Gauss1DsMAP, 205
 - mappel::Gauss1DsMLE, 234
 - mappel::Gauss1DsModel, 261
 - mappel::Gauss2DMAP, 289
 - mappel::Gauss2DMLE, 319
 - mappel::Gauss2DModel, 349
 - mappel::Gauss2DsMAP, 379
 - mappel::Gauss2DsMLE, 411
 - mappel::Gauss2DsModel, 443
 - mappel::Gauss2DsxyMAP, 472
 - mappel::Gauss2DsxyModel, 498
 - mappel::ImageFormat1DBase, 524
 - mappel::ImageFormat2DBase, 530
 - mappel::PoissonNoise1DObjective, 711
 - mappel::PoissonNoise2DObjective, 718

- get_stats
 - mappel::CGaussHeuristicEstimator, 92
 - mappel::CGaussMLE, 101
 - mappel::Estimator, 109
 - mappel::Gauss1DMAP, 123
 - mappel::Gauss1DMLE, 150
 - mappel::Gauss1DModel, 178
 - mappel::Gauss1DsMAP, 205
 - mappel::Gauss1DsMLE, 234
 - mappel::Gauss1DsModel, 261
 - mappel::Gauss2DMAP, 290
 - mappel::Gauss2DMLE, 320
 - mappel::Gauss2DModel, 349
 - mappel::Gauss2DsMAP, 379
 - mappel::Gauss2DsMLE, 411
 - mappel::Gauss2DsModel, 443
 - mappel::Gauss2DsxyMAP, 472

- mappel::Gauss2DsxyModel, [499](#)
- mappel::HeuristicEstimator, [517](#)
- mappel::ImageFormat1DBase, [524](#)
- mappel::ImageFormat2DBase, [530](#)
- mappel::IterativeMaximizer, [539](#)
- mappel::MAPEstimator, [551](#)
- mappel::MCMCAdaptor1Ds, [592](#)
- mappel::MCMCAdaptor1D, [574](#)
- mappel::MCMCAdaptor2Ds, [629](#)
- mappel::MCMCAdaptor2D, [610](#)
- mappel::MCMCAdaptorBase, [642](#)
- mappel::MLEstimator, [649](#)
- mappel::NewtonDiagonalMaximizer, [665](#)
- mappel::NewtonMaximizer, [678](#)
- mappel::PointEmitterModel, [696](#)
- mappel::PoissonNoise1DObjective, [711](#)
- mappel::PoissonNoise2DObjective, [718](#)
- mappel::QuasiNewtonMaximizer, [731](#)
- mappel::SimulatedAnnealingMaximizer, [742](#)
- mappel::ThreadedEstimator, [774](#)
- mappel::TrustRegionMaximizer, [786](#)
- get_theta_sequence
 - mappel::IterativeMaximizer::MaximizerData, [562](#)
- get_theta_sequence_rllh
 - mappel::IterativeMaximizer::MaximizerData, [562](#)
- get_ubound
 - mappel::Gauss1DMAP, [123](#)
 - mappel::Gauss1DMLE, [150](#)
 - mappel::Gauss1DModel, [178](#)
 - mappel::Gauss1DsMAP, [206](#)
 - mappel::Gauss1DsMLE, [234](#)
 - mappel::Gauss1DsModel, [261](#)
 - mappel::Gauss2DMAP, [290](#)
 - mappel::Gauss2DMLE, [320](#)
 - mappel::Gauss2DModel, [349](#)
 - mappel::Gauss2DsMAP, [379](#)
 - mappel::Gauss2DsMLE, [411](#)
 - mappel::Gauss2DsModel, [443](#)
 - mappel::Gauss2DsxyMAP, [472](#)
 - mappel::Gauss2DsxyModel, [499](#)
 - mappel::MAPEstimator, [551](#)
 - mappel::MCMCAdaptor1Ds, [592](#)
 - mappel::MCMCAdaptor1D, [574](#)
 - mappel::MCMCAdaptor2Ds, [629](#)
 - mappel::MCMCAdaptor2D, [610](#)
 - mappel::MLEstimator, [649](#)
 - mappel::PointEmitterModel, [696](#)
- getIteration
 - mappel::IterativeMaximizer::MaximizerData, [562](#)
- global_default_mcmc_sigma_scale
 - mappel::Gauss1DMAP, [135](#)
 - mappel::Gauss1DMLE, [163](#)
 - mappel::Gauss1DModel, [190](#)
 - mappel::Gauss1DsMAP, [219](#)
- mappel::Gauss1DsMLE, [247](#)
- mappel::Gauss1DsModel, [274](#)
- mappel::Gauss2DMAP, [303](#)
- mappel::Gauss2DMLE, [333](#)
- mappel::Gauss2DModel, [361](#)
- mappel::Gauss2DsMAP, [393](#)
- mappel::Gauss2DsMLE, [425](#)
- mappel::Gauss2DsModel, [456](#)
- mappel::Gauss2DsxyMAP, [484](#)
- mappel::MCMCAdaptor1Ds, [601](#)
- mappel::MCMCAdaptor1D, [583](#)
- mappel::MCMCAdaptor2Ds, [638](#)
- mappel::MCMCAdaptor2D, [619](#)
- mappel::MCMCAdaptorBase, [643](#)
- global_max_mcmc_sigma_scale
 - mappel::Gauss1DMAP, [135](#)
 - mappel::Gauss1DMLE, [163](#)
 - mappel::Gauss1DModel, [190](#)
 - mappel::Gauss1DsMAP, [219](#)
 - mappel::Gauss1DsMLE, [247](#)
 - mappel::Gauss1DsModel, [274](#)
 - mappel::Gauss2DMAP, [303](#)
 - mappel::Gauss2DMLE, [333](#)
 - mappel::Gauss2DModel, [362](#)
 - mappel::Gauss2DsMAP, [394](#)
 - mappel::Gauss2DsMLE, [426](#)
 - mappel::Gauss2DsModel, [456](#)
 - mappel::Gauss2DsxyMAP, [484](#)
 - mappel::MCMCAdaptor1Ds, [601](#)
 - mappel::MCMCAdaptor1D, [583](#)
 - mappel::MCMCAdaptor2Ds, [638](#)
 - mappel::MCMCAdaptor2D, [620](#)
 - mappel::MCMCAdaptorBase, [643](#)
- global_max_psf_sigma
 - mappel::Gauss1DMAP, [135](#)
 - mappel::Gauss1DMLE, [163](#)
 - mappel::Gauss1DModel, [190](#)
 - mappel::Gauss1DsMAP, [219](#)
 - mappel::Gauss1DsMLE, [247](#)
 - mappel::Gauss1DsModel, [274](#)
 - mappel::Gauss2DMAP, [303](#)
 - mappel::Gauss2DMLE, [333](#)
 - mappel::Gauss2DModel, [362](#)
 - mappel::Gauss2DsMAP, [394](#)
 - mappel::Gauss2DsMLE, [426](#)
 - mappel::Gauss2DsModel, [457](#)
 - mappel::Gauss2DsxyMAP, [484](#)
 - mappel::Gauss2DsxyModel, [508](#)
 - mappel::MAPEstimator, [558](#)
 - mappel::MCMCAdaptor1Ds, [601](#)
 - mappel::MCMCAdaptor1D, [583](#)
 - mappel::MCMCAdaptor2Ds, [638](#)
 - mappel::MCMCAdaptor2D, [620](#)
 - mappel::MLEstimator, [656](#)

- mappel::PointEmitterModel, 702
- global_max_size
 - mappel::Gauss1DModel, 136
 - mappel::Gauss1DModel, 164
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DsMAP, 219
 - mappel::Gauss1DsMLE, 247
 - mappel::Gauss1DsModel, 274
 - mappel::Gauss2DModel, 303
 - mappel::Gauss2DModel, 333
 - mappel::Gauss2DModel, 362
 - mappel::Gauss2DsMAP, 394
 - mappel::Gauss2DsMLE, 426
 - mappel::Gauss2DsModel, 457
 - mappel::Gauss2DsxyMAP, 485
 - mappel::Gauss2DsxyModel, 508
 - mappel::ImageFormat1DBase, 525
 - mappel::ImageFormat2DBase, 531
 - mappel::PoissonNoise1DObjective, 712
 - mappel::PoissonNoise2DObjective, 720
- global_min_psf_sigma
 - mappel::Gauss1DModel, 136
 - mappel::Gauss1DModel, 164
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DsMAP, 219
 - mappel::Gauss1DsMLE, 247
 - mappel::Gauss1DsModel, 274
 - mappel::Gauss2DModel, 303
 - mappel::Gauss2DModel, 333
 - mappel::Gauss2DModel, 362
 - mappel::Gauss2DsMAP, 394
 - mappel::Gauss2DsMLE, 426
 - mappel::Gauss2DsModel, 457
 - mappel::Gauss2DsxyMAP, 485
 - mappel::Gauss2DsxyModel, 508
 - mappel::MAPEstimator, 558
 - mappel::MCMCAdaptor1Ds, 602
 - mappel::MCMCAdaptor1D, 584
 - mappel::MCMCAdaptor2Ds, 638
 - mappel::MCMCAdaptor2D, 620
 - mappel::MLEstimator, 656
 - mappel::PointEmitterModel, 703
- global_min_size
 - mappel::Gauss1DModel, 136
 - mappel::Gauss1DModel, 164
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DsMAP, 219
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 275
 - mappel::Gauss2DModel, 303
 - mappel::Gauss2DModel, 333
 - mappel::Gauss2DModel, 362
 - mappel::Gauss2DsMAP, 394
 - mappel::Gauss2DsMLE, 426
- mappel::Gauss2DsModel, 457
- mappel::Gauss2DsxyMAP, 485
- mappel::Gauss2DsxyModel, 508
- mappel::ImageFormat1DBase, 525
- mappel::ImageFormat2DBase, 531
- mappel::PoissonNoise1DObjective, 713
- mappel::PoissonNoise2DObjective, 720
- grad
 - mappel::IterativeMaximizer::MaximizerData, 565
 - mappel::methods::likelihood, 63
 - mappel::methods::objective, 68, 69
- grad2
 - mappel::methods::likelihood, 63, 64
 - mappel::methods::objective, 69
- grad_components
 - mappel::methods::likelihood::debug, 66
 - mappel::methods::objective::debug, 73
- grad_stack
 - mappel::methods::objective::openmp, 75
- GradRatio
 - mappel::CGaussHeuristicEstimator, 89
 - mappel::CGaussMLE, 97
 - mappel::Estimator, 105
 - mappel::HeuristicEstimator, 513
 - mappel::IterativeMaximizer, 535
 - mappel::NewtonDiagonalMaximizer, 661
 - mappel::NewtonMaximizer, 674
 - mappel::QuasiNewtonMaximizer, 726
 - mappel::SimulatedAnnealingMaximizer, 739
 - mappel::ThreadedEstimator, 771
 - mappel::TrustRegionMaximizer, 780
- Gx
 - mappel::Gauss1DModel::Stencil, 748
 - mappel::Gauss1DsModel::Stencil, 752
 - mappel::Gauss2DModel::Stencil, 768
 - mappel::Gauss2DsModel::Stencil, 763
 - mappel::Gauss2DsxyModel::Stencil, 757
- Gy
 - mappel::Gauss2DModel::Stencil, 768
 - mappel::Gauss2DsModel::Stencil, 763
 - mappel::Gauss2DsxyModel::Stencil, 757
- has_hyperparam
 - mappel::Gauss1DModel, 123
 - mappel::Gauss1DModel, 151
 - mappel::Gauss1DModel, 178
 - mappel::Gauss1DsMAP, 206
 - mappel::Gauss1DsMLE, 234
 - mappel::Gauss1DsModel, 261
 - mappel::Gauss2DModel, 290
 - mappel::Gauss2DModel, 320
 - mappel::Gauss2DModel, 349
 - mappel::Gauss2DsMAP, 379
 - mappel::Gauss2DsMLE, 411

- mappel::Gauss2DsModel, [443](#)
- mappel::Gauss2DsxyMAP, [472](#)
- mappel::Gauss2DsxyModel, [499](#)
- mappel::MAPEstimator, [551](#)
- mappel::MCMCAdaptor1Ds, [593](#)
- mappel::MCMCAdaptor1D, [575](#)
- mappel::MCMCAdaptor2Ds, [629](#)
- mappel::MCMCAdaptor2D, [611](#)
- mappel::MLEstimator, [649](#)
- mappel::PointEmitterModel, [696](#)
- hessian
 - mappel::methods::likelihood, [64](#)
 - mappel::methods::objective, [69](#), [70](#)
- hessian_components
 - mappel::methods::likelihood::debug, [66](#)
 - mappel::methods::objective::debug, [73](#)
- hessian_stack
 - mappel::methods::objective::openmp, [76](#)
- HeuristicEstimator
 - mappel::HeuristicEstimator, [513](#)
- I
 - mappel::Gauss1DModel::Stencil, [747](#)
 - mappel::Gauss1DsModel::Stencil, [751](#)
 - mappel::Gauss2DModel::Stencil, [766](#)
 - mappel::Gauss2DsModel::Stencil, [760](#)
 - mappel::Gauss2DsxyModel::Stencil, [755](#)
- icontains
 - mappel, [35](#)
- IdxMatT
 - mappel, [25](#)
- IdxVecT
 - mappel, [25](#)
- IdxT
 - mappel, [25](#)
- im
 - mappel::IterativeMaximizer::MaximizerData, [565](#)
- ImageCoordT
 - mappel, [25](#)
 - mappel::Gauss1DMAP, [116](#)
 - mappel::Gauss1DMLE, [143](#)
 - mappel::Gauss1DModel, [171](#)
 - mappel::Gauss1DsMAP, [198](#)
 - mappel::Gauss1DsMLE, [227](#)
 - mappel::Gauss1DsModel, [254](#)
 - mappel::Gauss2DMAP, [282](#)
 - mappel::Gauss2DMLE, [312](#)
 - mappel::Gauss2DModel, [341](#)
 - mappel::Gauss2DsMAP, [370](#)
 - mappel::Gauss2DsMLE, [402](#)
 - mappel::Gauss2DsModel, [435](#)
 - mappel::Gauss2DsxyMAP, [465](#)
 - mappel::Gauss2DsxyModel, [492](#)
 - mappel::ImageFormat1DBase, [521](#)
- mappel::ImageFormat2DBase, [527](#)
- mappel::PoissonNoise1DObjective, [708](#)
- mappel::PoissonNoise2DObjective, [715](#)
- ImageFormat1DBase
 - mappel::ImageFormat1DBase, [522](#)
- ImageFormat1DBase.cpp, [814](#)
- ImageFormat1DBase.h, [815](#)
- ImageFormat2DBase
 - mappel::ImageFormat2DBase, [529](#)
- ImageFormat2DBase.cpp, [816](#)
- ImageFormat2DBase.h, [816](#)
- ImagePixelT
 - mappel, [26](#)
 - mappel::Gauss1DMAP, [116](#)
 - mappel::Gauss1DMLE, [143](#)
 - mappel::Gauss1DModel, [171](#)
 - mappel::Gauss1DsMAP, [198](#)
 - mappel::Gauss1DsMLE, [227](#)
 - mappel::Gauss1DsModel, [254](#)
 - mappel::Gauss2DMAP, [282](#)
 - mappel::Gauss2DMLE, [312](#)
 - mappel::Gauss2DModel, [342](#)
 - mappel::Gauss2DsMAP, [371](#)
 - mappel::Gauss2DsMLE, [403](#)
 - mappel::Gauss2DsModel, [435](#)
 - mappel::Gauss2DsxyMAP, [465](#)
 - mappel::Gauss2DsxyModel, [492](#)
 - mappel::ImageFormat1DBase, [521](#)
 - mappel::ImageFormat2DBase, [527](#)
 - mappel::PoissonNoise1DObjective, [708](#)
 - mappel::PoissonNoise2DObjective, [715](#)
- ImageShapeT
 - mappel::Gauss1DMAP, [116](#)
 - mappel::Gauss1DMLE, [143](#)
 - mappel::Gauss1DModel, [171](#)
 - mappel::Gauss1DsMAP, [198](#)
 - mappel::Gauss1DsMLE, [227](#)
 - mappel::Gauss1DsModel, [255](#)
 - mappel::Gauss2DMAP, [282](#)
 - mappel::Gauss2DMLE, [312](#)
 - mappel::Gauss2DModel, [342](#)
 - mappel::Gauss2DsMAP, [371](#)
 - mappel::Gauss2DsMLE, [403](#)
 - mappel::Gauss2DsModel, [435](#)
 - mappel::Gauss2DsxyMAP, [465](#)
 - mappel::Gauss2DsxyModel, [492](#)
 - mappel::ImageFormat1DBase, [521](#)
 - mappel::ImageFormat2DBase, [527](#)
 - mappel::PoissonNoise1DObjective, [708](#)
 - mappel::PoissonNoise2DObjective, [715](#)
- ImageSizeShapeT
 - mappel::Gauss1DMAP, [116](#)
 - mappel::Gauss1DMLE, [143](#)
 - mappel::Gauss1DModel, [172](#)

- mappel::Gauss1DsMAP, 198
- mappel::Gauss1DsMLE, 227
- mappel::Gauss1DsModel, 255
- mappel::Gauss2DMAP, 282
- mappel::Gauss2DMLE, 312
- mappel::Gauss2DModel, 342
- mappel::Gauss2DsMAP, 371
- mappel::Gauss2DsMLE, 403
- mappel::Gauss2DsModel, 435
- mappel::Gauss2DsxyMAP, 465
- mappel::Gauss2DsxyModel, 492
- mappel::ImageFormat1DBase, 521
- mappel::ImageFormat2DBase, 528
- mappel::PoissonNoise1DObjective, 708
- mappel::PoissonNoise2DObjective, 716
- ImageSizeVecShapeT
 - mappel::Gauss1DMAP, 116
 - mappel::Gauss1DMLE, 144
 - mappel::Gauss1DModel, 172
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 255
 - mappel::Gauss2DMAP, 282
 - mappel::Gauss2DMLE, 312
 - mappel::Gauss2DModel, 342
 - mappel::Gauss2DsMAP, 371
 - mappel::Gauss2DsMLE, 403
 - mappel::Gauss2DsModel, 435
 - mappel::Gauss2DsxyMAP, 466
 - mappel::Gauss2DsxyModel, 492
 - mappel::ImageFormat1DBase, 522
 - mappel::ImageFormat2DBase, 528
 - mappel::PoissonNoise1DObjective, 709
 - mappel::PoissonNoise2DObjective, 716
- ImageSizeVecT
 - mappel::Gauss1DMAP, 116
 - mappel::Gauss1DMLE, 144
 - mappel::Gauss1DModel, 172
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 255
 - mappel::Gauss2DMAP, 283
 - mappel::Gauss2DMLE, 313
 - mappel::Gauss2DModel, 342
 - mappel::Gauss2DsMAP, 371
 - mappel::Gauss2DsMLE, 403
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyMAP, 466
 - mappel::Gauss2DsxyModel, 493
 - mappel::ImageFormat1DBase, 522
 - mappel::ImageFormat2DBase, 528
 - mappel::PoissonNoise1DObjective, 709
 - mappel::PoissonNoise2DObjective, 716
- ImageSizeT
 - mappel::Gauss1DMAP, 116
 - mappel::Gauss1DMLE, 144
 - mappel::Gauss1DModel, 172
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 255
 - mappel::Gauss2DMAP, 283
 - mappel::Gauss2DMLE, 313
 - mappel::Gauss2DModel, 343
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyMAP, 466
 - mappel::Gauss2DsxyModel, 493
 - mappel::ImageFormat1DBase, 522
- mappel::Gauss1DMAP, 116
- mappel::Gauss1DMLE, 144
- mappel::Gauss1DModel, 172
- mappel::Gauss1DsMAP, 199
- mappel::Gauss1DsMLE, 227
- mappel::Gauss1DsModel, 255
- mappel::Gauss2DMAP, 282
- mappel::Gauss2DMLE, 312
- mappel::Gauss2DModel, 342
- mappel::Gauss2DsMAP, 371
- mappel::Gauss2DsMLE, 403
- mappel::Gauss2DsModel, 435
- mappel::Gauss2DsxyMAP, 466
- mappel::Gauss2DsxyModel, 492
- mappel::ImageFormat1DBase, 521
- mappel::ImageFormat2DBase, 528
- mappel::PoissonNoise1DObjective, 708
- mappel::PoissonNoise2DObjective, 716
- ImageStackShapeT
 - mappel::Gauss1DMAP, 116
 - mappel::Gauss1DMLE, 144
 - mappel::Gauss1DModel, 172
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 255
 - mappel::Gauss2DMAP, 283
 - mappel::Gauss2DMLE, 313
 - mappel::Gauss2DModel, 342
 - mappel::Gauss2DsMAP, 371
 - mappel::Gauss2DsMLE, 403
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyMAP, 466
 - mappel::Gauss2DsxyModel, 493
 - mappel::ImageFormat1DBase, 522
 - mappel::ImageFormat2DBase, 528
 - mappel::PoissonNoise1DObjective, 709
 - mappel::PoissonNoise2DObjective, 716
- ImageStackT
 - mappel, 26
 - mappel::Gauss1DMAP, 117
 - mappel::Gauss1DMLE, 144
 - mappel::Gauss1DModel, 172
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 228
 - mappel::Gauss1DsModel, 255
 - mappel::Gauss2DMAP, 283
 - mappel::Gauss2DMLE, 313
 - mappel::Gauss2DModel, 343
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyMAP, 466
 - mappel::Gauss2DsxyModel, 493
 - mappel::ImageFormat1DBase, 522

- mappel::ImageFormat2DBase, 528
- mappel::PoissonNoise1DObjective, 709
- mappel::PoissonNoise2DObjective, 716
- ImageT
 - mappel, 26
 - mappel::Gauss1DMap, 117
 - mappel::Gauss1DMLE, 144
 - mappel::Gauss1DModel, 172
 - mappel::Gauss1DsMap, 199
 - mappel::Gauss1DsMLE, 228
 - mappel::Gauss1DsModel, 256
 - mappel::Gauss2DMap, 283
 - mappel::Gauss2DMLE, 313
 - mappel::Gauss2DModel, 343
 - mappel::Gauss2DsMap, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyMap, 466
 - mappel::Gauss2DsxyModel, 493
 - mappel::ImageFormat1DBase, 522
 - mappel::ImageFormat2DBase, 528
 - mappel::PoissonNoise1DObjective, 709
 - mappel::PoissonNoise2DObjective, 716
- initial_theta_estimate
 - mappel::Gauss1DMap, 123
 - mappel::Gauss1DMLE, 151
 - mappel::Gauss1DModel, 178, 179
 - mappel::Gauss1DsMap, 206
 - mappel::Gauss1DsMLE, 234
 - mappel::Gauss1DsModel, 262
 - mappel::Gauss2DMap, 290
 - mappel::Gauss2DMLE, 320
 - mappel::Gauss2DModel, 349, 350
 - mappel::Gauss2DsMap, 379, 380
 - mappel::Gauss2DsMLE, 411, 412
 - mappel::Gauss2DsModel, 443, 444
 - mappel::Gauss2DsxyMap, 472, 473
 - mappel::Gauss2DsxyModel, 499
- Install.md, 817
- is_negative_definite
 - mappel, 35
- is_positive_definite
 - mappel, 35
- is_symmetric
 - mappel, 35
- istarts_with
 - mappel, 35, 36
- IterativeMaximizer
 - mappel::IterativeMaximizer, 535
- lambda_min
 - mappel::IterativeMaximizer, 542
 - mappel::NewtonDiagonalMaximizer, 669
 - mappel::NewtonMaximizer, 682
 - mappel::QuasiNewtonMaximizer, 734
 - mappel::TrustRegionMaximizer, 790
- lambda_term_color
 - mappel, 36
- last_backtrack_idx
 - mappel::IterativeMaximizer, 542
 - mappel::NewtonDiagonalMaximizer, 669
 - mappel::NewtonMaximizer, 682
 - mappel::QuasiNewtonMaximizer, 734
 - mappel::TrustRegionMaximizer, 790
- lbound
 - mappel::Gauss1DMap, 136
 - mappel::Gauss1DMLE, 164
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DsMap, 220
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 275
 - mappel::Gauss2DMap, 304
 - mappel::Gauss2DMLE, 334
 - mappel::Gauss2DModel, 362
 - mappel::Gauss2DsMap, 394
 - mappel::Gauss2DsMLE, 426
 - mappel::Gauss2DsModel, 457
 - mappel::Gauss2DsxyMap, 485
 - mappel::Gauss2DsxyModel, 508
 - mappel::IterativeMaximizer::MaximizerData, 565
 - mappel::MAPEstimator, 559
 - mappel::MCMCAdaptor1Ds, 602
 - mappel::MCMCAdaptor1D, 584
 - mappel::MCMCAdaptor2Ds, 639
 - mappel::MCMCAdaptor2D, 620
 - mappel::MLEstimator, 657
 - mappel::PointEmitterModel, 703
- likelihood_objective
 - mappel::methods, 57
- llh
 - mappel::methods::likelihood, 64
 - mappel::methods::objective, 70, 71
- llh_components
 - mappel::methods::likelihood::debug, 66
 - mappel::methods::objective::debug, 74
- llh_stack
 - mappel::methods::objective::openmp, 76, 77
- local_maximize
 - mappel::IterativeMaximizer, 539
 - mappel::NewtonDiagonalMaximizer, 666
 - mappel::NewtonMaximizer, 679
 - mappel::QuasiNewtonMaximizer, 731
 - mappel::TrustRegionMaximizer, 786
- log_likelihood
 - PoissonGaussianNoise2DObjective.h, 842
- log_likelihood_stack
 - mappel, 36
- log_prior_l_const

- mappel::PriorMAP1DObjective, 723
- log_prior_beta2_const
 - mappel, 37
- log_prior_beta_const
 - mappel, 37
- log_prior_bg_const
 - mappel::PriorMAP1DObjective, 723
- log_prior_gamma_const
 - mappel, 37
- log_prior_normal_const
 - mappel, 37
- log_prior_pareto_const
 - mappel, 37
- log_prior_pos_const
 - mappel::PriorMAP1DObjective, 723
- LogicalError
 - mappel::LogicalError, 545
- MAPEstimator
 - mappel::MAPEstimator, 548
- MAPEstimator.h, 817
- MCMCAdaptor1D.cpp, 820
- MCMCAdaptor1D.h, 820
- MCMCAdaptor1Ds
 - mappel::MCMCAdaptor1Ds, 589
- MCMCAdaptor1Ds.cpp, 821
- MCMCAdaptor1Ds.h, 822
- MCMCAdaptor1D
 - mappel::MCMCAdaptor1D, 571
- MCMCAdaptor2D.cpp, 822
- MCMCAdaptor2D.h, 823
- MCMCAdaptor2Ds
 - mappel::MCMCAdaptor2Ds, 625, 626
- MCMCAdaptor2Ds.cpp, 823
- MCMCAdaptor2Ds.h, 824
- MCMCAdaptor2D
 - mappel::MCMCAdaptor2D, 607
- MCMCAdaptorBase
 - mappel::MCMCAdaptorBase, 641
- MCMCAdaptorBase.cpp, 824
- MCMCAdaptorBase.h, 825
- MLEstimator
 - mappel::MLEstimator, 646
- MLEstimator.h, 825
- make_DX_stencil
 - mappel, 38
- make_DXS2_stencil
 - mappel, 38
- make_DXS_stencil
 - mappel, 38
- make_DXSX_stencil
 - mappel, 38
- make_G_stencil
 - mappel, 38
- make_X_stencil
 - mappel, 39
- make_d_stencil
 - mappel, 37
- make_default_prior
 - mappel::Gauss1DMap, 123
 - mappel::Gauss1DMLE, 151
 - mappel::Gauss1DModel, 179
 - mappel::Gauss1DsMAP, 206
 - mappel::Gauss1DsMLE, 234
 - mappel::Gauss1DsModel, 262
 - mappel::Gauss2DMap, 291
 - mappel::Gauss2DMLE, 321
 - mappel::Gauss2DModel, 350
 - mappel::Gauss2DsMAP, 380
 - mappel::Gauss2DsMLE, 412
 - mappel::Gauss2DsModel, 444
 - mappel::Gauss2DsxyMAP, 473
 - mappel::Gauss2DsxyModel, 499
- make_default_prior_beta_position
 - mappel::Gauss1DMap, 124
 - mappel::Gauss1DMLE, 151
 - mappel::Gauss1DModel, 179
 - mappel::Gauss1DsMAP, 206
 - mappel::Gauss1DsMLE, 235
 - mappel::Gauss1DsModel, 262
 - mappel::Gauss2DMap, 291
 - mappel::Gauss2DMLE, 321
 - mappel::Gauss2DModel, 350
 - mappel::Gauss2DsMAP, 380
 - mappel::Gauss2DsMLE, 412
 - mappel::Gauss2DsModel, 444
 - mappel::Gauss2DsxyMAP, 473
- make_default_prior_normal_position
 - mappel::Gauss1DMap, 124
 - mappel::Gauss1DMLE, 151
 - mappel::Gauss1DModel, 179
 - mappel::Gauss1DsMAP, 207
 - mappel::Gauss1DsMLE, 235
 - mappel::Gauss1DsModel, 262
 - mappel::Gauss2DMap, 291
 - mappel::Gauss2DMLE, 321
 - mappel::Gauss2DModel, 350
 - mappel::Gauss2DsMAP, 380
 - mappel::Gauss2DsMLE, 412
 - mappel::Gauss2DsModel, 444
 - mappel::Gauss2DsxyMAP, 473
- make_estimator
 - mappel::methods, 57
 - PoissonGaussianNoise2DObjective.h, 842
- make_gaussian_stencil
 - mappel, 39
- make_image
 - mappel::Gauss1DMap, 124

- mappel::Gauss1DMLE, [152](#)
- mappel::Gauss1DModel, [179](#)
- mappel::Gauss1DsMAP, [207](#)
- mappel::Gauss1DsMLE, [235](#)
- mappel::Gauss1DsModel, [262](#)
- mappel::Gauss2DMAP, [291](#)
- mappel::Gauss2DMLE, [321](#)
- mappel::Gauss2DModel, [350](#)
- mappel::Gauss2DsMAP, [381](#)
- mappel::Gauss2DsMLE, [413](#)
- mappel::Gauss2DsModel, [444](#)
- mappel::Gauss2DsxyMAP, [473](#)
- mappel::Gauss2DsxyModel, [499](#)
- mappel::ImageFormat1DBase, [524](#)
- mappel::ImageFormat2DBase, [530](#)
- mappel::PoissonNoise1DObjective, [711](#)
- mappel::PoissonNoise2DObjective, [719](#)
- make_image_stack
 - mappel::Gauss1DMAP, [124](#)
 - mappel::Gauss1DMLE, [152](#)
 - mappel::Gauss1DModel, [179](#)
 - mappel::Gauss1DsMAP, [207](#)
 - mappel::Gauss1DsMLE, [235](#)
 - mappel::Gauss1DsModel, [263](#)
 - mappel::Gauss2DMAP, [291](#)
 - mappel::Gauss2DMLE, [321](#)
 - mappel::Gauss2DModel, [350](#)
 - mappel::Gauss2DsMAP, [381](#)
 - mappel::Gauss2DsMLE, [413](#)
 - mappel::Gauss2DsModel, [445](#)
 - mappel::Gauss2DsxyMAP, [473](#)
 - mappel::Gauss2DsxyModel, [499](#)
 - mappel::ImageFormat1DBase, [524](#)
 - mappel::ImageFormat2DBase, [530](#)
 - mappel::PoissonNoise1DObjective, [711](#)
 - mappel::PoissonNoise2DObjective, [719](#)
- make_internal_1Dsum_estimator
 - mappel::Gauss2DMAP, [291](#)
 - mappel::Gauss2DMLE, [321](#)
 - mappel::Gauss2DModel, [351](#)
 - mappel::Gauss2DsMAP, [381](#)
 - mappel::Gauss2DsMLE, [413](#)
 - mappel::Gauss2DsModel, [445](#)
- make_param
 - mappel::Gauss1DMAP, [124](#)
 - mappel::Gauss1DMLE, [152](#)
 - mappel::Gauss1DModel, [179](#), [180](#)
 - mappel::Gauss1DsMAP, [207](#)
 - mappel::Gauss1DsMLE, [235](#)
 - mappel::Gauss1DsModel, [263](#)
 - mappel::Gauss2DMAP, [292](#)
 - mappel::Gauss2DMLE, [322](#)
 - mappel::Gauss2DModel, [351](#)
 - mappel::Gauss2DsMAP, [381](#)
- mappel::Gauss2DsMLE, [413](#)
- mappel::Gauss2DsModel, [443](#)
- mappel::Gauss2DsModel, [445](#)
- mappel::Gauss2DsxyMAP, [474](#)
- mappel::Gauss2DsxyModel, [500](#)
- mappel::MAPEstimator, [552](#)
- mappel::MCMCAdaptor1Ds, [593](#)
- mappel::MCMCAdaptor1D, [575](#)
- mappel::MCMCAdaptor2Ds, [629](#), [630](#)
- mappel::MCMCAdaptor2D, [611](#)
- mappel::MLEstimator, [650](#)
- mappel::PointEmitterModel, [696](#)
- make_param_mat
 - mappel::Gauss1DMAP, [125](#)
 - mappel::Gauss1DMLE, [152](#)
 - mappel::Gauss1DModel, [180](#)
 - mappel::Gauss1DsMAP, [207](#), [208](#)
 - mappel::Gauss1DsMLE, [236](#)
 - mappel::Gauss1DsModel, [263](#)
 - mappel::Gauss2DMAP, [292](#)
 - mappel::Gauss2DMLE, [322](#)
 - mappel::Gauss2DModel, [351](#)
 - mappel::Gauss2DsMAP, [382](#)
 - mappel::Gauss2DsMLE, [414](#)
 - mappel::Gauss2DsModel, [445](#)
 - mappel::Gauss2DsxyMAP, [474](#)
 - mappel::Gauss2DsxyModel, [500](#)
 - mappel::MAPEstimator, [552](#)
 - mappel::MCMCAdaptor1Ds, [593](#)
 - mappel::MCMCAdaptor1D, [575](#)
 - mappel::MCMCAdaptor2Ds, [630](#)
 - mappel::MCMCAdaptor2D, [611](#)
 - mappel::MLEstimator, [650](#)
 - mappel::PointEmitterModel, [696](#), [697](#)
- make_param_mat_stack
 - mappel::Gauss1DMAP, [125](#)
 - mappel::Gauss1DMLE, [153](#)
 - mappel::Gauss1DModel, [180](#)
 - mappel::Gauss1DsMAP, [208](#)
 - mappel::Gauss1DsMLE, [236](#)
 - mappel::Gauss1DsModel, [263](#)
 - mappel::Gauss2DMAP, [292](#)
 - mappel::Gauss2DMLE, [322](#)
 - mappel::Gauss2DModel, [351](#), [352](#)
 - mappel::Gauss2DsMAP, [382](#)
 - mappel::Gauss2DsMLE, [414](#)
 - mappel::Gauss2DsModel, [446](#)
 - mappel::Gauss2DsxyMAP, [474](#)
 - mappel::Gauss2DsxyModel, [500](#)
 - mappel::MAPEstimator, [552](#)
 - mappel::MCMCAdaptor1Ds, [593](#), [594](#)
 - mappel::MCMCAdaptor1D, [575](#), [576](#)
 - mappel::MCMCAdaptor2Ds, [630](#)
 - mappel::MCMCAdaptor2D, [611](#), [612](#)
 - mappel::MLEstimator, [650](#)

- mappel::PointEmitterModel, 697
- make_param_stack
 - mappel::Gauss1DMAP, 125
 - mappel::Gauss1DMLE, 153
 - mappel::Gauss1DModel, 180
 - mappel::Gauss1DsMAP, 208
 - mappel::Gauss1DsMLE, 236
 - mappel::Gauss1DsModel, 264
 - mappel::Gauss2DMAP, 293
 - mappel::Gauss2DMLE, 323
 - mappel::Gauss2DModel, 352
 - mappel::Gauss2DsMAP, 382
 - mappel::Gauss2DsMLE, 414
 - mappel::Gauss2DsModel, 446
 - mappel::Gauss2DsxyMAP, 474, 475
 - mappel::Gauss2DsxyModel, 500, 501
 - mappel::MAPEstimator, 552, 553
 - mappel::MCMCAdaptor1Ds, 594
 - mappel::MCMCAdaptor1D, 576
 - mappel::MCMCAdaptor2Ds, 630
 - mappel::MCMCAdaptor2D, 612
 - mappel::MLEstimator, 650, 651
 - mappel::PointEmitterModel, 697
- make_prior_beta_position
 - mappel::Gauss1DMAP, 125
 - mappel::Gauss1DMLE, 153
 - mappel::Gauss1DModel, 181
 - mappel::Gauss1DsMAP, 208
 - mappel::Gauss1DsMLE, 236
 - mappel::Gauss1DsModel, 264
 - mappel::Gauss2DMAP, 293
 - mappel::Gauss2DMLE, 323
 - mappel::Gauss2DModel, 352
 - mappel::Gauss2DsMAP, 382
 - mappel::Gauss2DsMLE, 414
 - mappel::Gauss2DsModel, 446
 - mappel::Gauss2DsxyMAP, 475
 - mappel::Gauss2DsxyModel, 501
- make_prior_component_intensity
 - mappel::Gauss1DMAP, 126
 - mappel::Gauss1DMLE, 153
 - mappel::Gauss1DModel, 181
 - mappel::Gauss1DsMAP, 208
 - mappel::Gauss1DsMLE, 237
 - mappel::Gauss1DsModel, 264
 - mappel::Gauss2DMAP, 293
 - mappel::Gauss2DMLE, 323
 - mappel::Gauss2DModel, 352
 - mappel::Gauss2DsMAP, 383
 - mappel::Gauss2DsMLE, 415
 - mappel::Gauss2DsModel, 446
 - mappel::Gauss2DsxyMAP, 475
 - mappel::Gauss2DsxyModel, 501
- make_prior_component_position_beta
 - mappel::Gauss1DMAP, 126
 - mappel::Gauss1DMLE, 154
 - mappel::Gauss1DModel, 181
 - mappel::Gauss1DsMAP, 209
 - mappel::Gauss1DsMLE, 237
 - mappel::Gauss1DsModel, 264
 - mappel::Gauss2DMAP, 293
 - mappel::Gauss2DMLE, 323
 - mappel::Gauss2DModel, 352
 - mappel::Gauss2DsMAP, 383
 - mappel::Gauss2DsMLE, 415
 - mappel::Gauss2DsModel, 447
 - mappel::Gauss2DsxyMAP, 475
 - mappel::Gauss2DsxyModel, 501
 - mappel::MAPEstimator, 553
 - mappel::MCMCAdaptor1Ds, 594
 - mappel::MCMCAdaptor1D, 576
 - mappel::MCMCAdaptor2Ds, 631
 - mappel::MCMCAdaptor2D, 612
 - mappel::MLEstimator, 651
 - mappel::PointEmitterModel, 697
- make_prior_component_position_normal
 - mappel::Gauss1DMAP, 126
 - mappel::Gauss1DMLE, 154
 - mappel::Gauss1DModel, 181
 - mappel::Gauss1DsMAP, 209
 - mappel::Gauss1DsMLE, 237
 - mappel::Gauss1DsModel, 265
 - mappel::Gauss2DMAP, 294
 - mappel::Gauss2DMLE, 324
 - mappel::Gauss2DModel, 353
 - mappel::Gauss2DsMAP, 383
 - mappel::Gauss2DsMLE, 415
 - mappel::Gauss2DsModel, 447
 - mappel::Gauss2DsxyMAP, 475
 - mappel::Gauss2DsxyModel, 501
 - mappel::MAPEstimator, 553
 - mappel::MCMCAdaptor1Ds, 595
 - mappel::MCMCAdaptor1D, 577
 - mappel::MCMCAdaptor2Ds, 631
 - mappel::MCMCAdaptor2D, 613
 - mappel::MLEstimator, 651
 - mappel::PointEmitterModel, 698
- make_prior_component_sigma
 - mappel::Gauss1DMAP, 126
 - mappel::Gauss1DMLE, 154
 - mappel::Gauss1DModel, 181

- mappel::Gauss1DsMAP, [209](#)
- mappel::Gauss1DsMLE, [237](#)
- mappel::Gauss1DsModel, [265](#)
- mappel::Gauss2DMAP, [294](#)
- mappel::Gauss2DMLE, [324](#)
- mappel::Gauss2DModel, [353](#)
- mappel::Gauss2DsMAP, [383](#)
- mappel::Gauss2DsMLE, [415](#)
- mappel::Gauss2DsModel, [447](#)
- mappel::Gauss2DsxyMAP, [476](#)
- mappel::Gauss2DsxyModel, [502](#)
- mappel::MAPEstimator, [553](#)
- mappel::MCMCAdaptor1Ds, [595](#)
- mappel::MCMCAdaptor1D, [577](#)
- mappel::MCMCAdaptor2Ds, [631](#)
- mappel::MCMCAdaptor2D, [613](#)
- mappel::MLEstimator, [651](#)
- mappel::PointEmitterModel, [698](#)
- make_prior_normal_position
 - mappel::Gauss1DMAP, [127](#)
 - mappel::Gauss1DMLE, [154](#)
 - mappel::Gauss1DModel, [182](#)
 - mappel::Gauss1DsMAP, [209](#)
 - mappel::Gauss1DsMLE, [238](#)
 - mappel::Gauss1DsModel, [265](#)
 - mappel::Gauss2DMAP, [294](#)
 - mappel::Gauss2DMLE, [324](#)
 - mappel::Gauss2DModel, [353](#)
 - mappel::Gauss2DsMAP, [384](#)
 - mappel::Gauss2DsMLE, [416](#)
 - mappel::Gauss2DsModel, [447](#)
 - mappel::Gauss2DsxyMAP, [476](#)
 - mappel::Gauss2DsxyModel, [502](#)
- make_stencil
 - mappel::Gauss1DMAP, [127](#)
 - mappel::Gauss1DMLE, [155](#)
 - mappel::Gauss1DModel, [182](#)
 - mappel::Gauss1DsMAP, [210](#)
 - mappel::Gauss1DsMLE, [238](#)
 - mappel::Gauss1DsModel, [265](#)
 - mappel::Gauss2DMAP, [294](#)
 - mappel::Gauss2DMLE, [324](#)
 - mappel::Gauss2DModel, [353](#)
 - mappel::Gauss2DsMAP, [384](#)
 - mappel::Gauss2DsMLE, [416](#)
 - mappel::Gauss2DsModel, [447](#)
 - mappel::Gauss2DsxyMAP, [476](#)
 - mappel::Gauss2DsxyModel, [502](#)
- make_unique
 - mappel, [39](#)
- mappel, [17](#)
 - beta2_prior_grad, [28](#)
 - beta2_prior_grad2, [28](#)
 - beta_prior_grad, [28](#)
 - beta_prior_grad2, [28](#)
 - BoolVecT, [25](#)
 - BoolT, [25](#)
 - cgauss_compute_estimate, [28](#)
 - cgauss_compute_estimate_debug, [28](#), [29](#)
 - cgauss_heuristic_compute_estimate, [29](#)
 - check_lower_bound_hyperparameter, [29](#)
 - check_positive_hyperparameter, [29](#)
 - check_unit_hyperparameter, [29](#)
 - cholesky, [29](#)
 - cholesky_convert_full_matrix, [30](#)
 - cholesky_convert_lower_triangular, [30](#)
 - cholesky_make_negative_definite, [30](#)
 - cholesky_make_positive_definite, [30](#)
 - cholesky_solve, [30](#)
 - copy_Lsym_mat, [31](#)
 - copy_Usym_mat, [31](#)
 - copy_Usym_mat_stack, [31](#)
 - cr_lower_bound_stack, [31](#)
 - CubeT, [25](#)
 - enable_all_cpus, [31](#)
 - EnableIfSubclassT, [25](#)
 - estimate_background, [31](#)
 - estimate_gaussian_2Dmax, [32](#)
 - estimate_gaussian_3Dmax, [32](#)
 - estimate_intensity, [32](#)
 - fill_DX_stencil, [32](#)
 - fill_DXS2_stencil, [32](#)
 - fill_DXS_stencil, [32](#)
 - fill_DXSX_stencil, [33](#)
 - fill_G_stencil, [33](#)
 - fill_X_stencil, [33](#)
 - fill_d_stencil, [32](#)
 - fill_gaussian_stencil, [33](#)
 - fisher_information_stack, [33](#)
 - gamma_prior_grad, [33](#)
 - gamma_prior_grad2, [33](#)
 - gauss_norm, [34](#)
 - gaussian_3D_convolution, [34](#)
 - gaussian_convolution, [34](#)
 - generate_poisson, [34](#)
 - generate_poisson_large, [34](#)
 - generate_poisson_small, [34](#)
 - icontains, [35](#)
 - IdxMatT, [25](#)
 - IdxVecT, [25](#)
 - IdxT, [25](#)
 - ImageCoordT, [25](#)
 - ImagePixelT, [26](#)
 - ImageStackT, [26](#)
 - ImageT, [26](#)
 - is_negative_definite, [35](#)
 - is_positive_definite, [35](#)
 - is_symmetric, [35](#)

istarts_with, [35](#), [36](#)
lambda_term_color, [36](#)
log_likelihood_stack, [36](#)
log_prior_beta2_const, [37](#)
log_prior_beta_const, [37](#)
log_prior_gamma_const, [37](#)
log_prior_normal_const, [37](#)
log_prior_pareto_const, [37](#)
make_DX_stencil, [38](#)
make_DXS2_stencil, [38](#)
make_DXS_stencil, [38](#)
make_DXSX_stencil, [38](#)
make_G_stencil, [38](#)
make_X_stencil, [39](#)
make_d_stencil, [37](#)
make_gaussian_stencil, [39](#)
make_unique, [39](#)
MappelError, [26](#)
MatT, [26](#)
maxidx, [39](#)
model_grad_stack, [39](#)
model_hessian_stack, [40](#)
model_image_stack, [40](#)
model_positive_hessian_stack, [41](#)
ModelDataStackT, [26](#)
ModelDataT, [26](#)
modified_cholesky, [41](#)
normal_prior_grad, [42](#)
normal_prior_grad2, [42](#)
normal_quantile_onesided, [42](#)
normal_quantile_twosided, [42](#)
operator<<, [42](#), [43](#)
ParallelRngGeneratorT, [26](#)
ParallelRngManagerT, [26](#)
ParamVecT, [26](#)
ParamT, [26](#)
pareto_prior_grad, [43](#)
pareto_prior_grad2, [43](#)
poisson_log_likelihood, [43](#)
print_centered_title, [43](#)
print_image, [44](#)
print_labeled_image, [44](#)
print_vec_row, [44](#)
refine_gaussian_2Dmax, [44](#)
refine_gaussian_3Dmax, [44](#)
relative_poisson_log_likelihood, [45](#)
restrict_value_range, [45](#)
ReturnIfSubclassT, [27](#)
rllh_beta2_prior, [45](#)
rllh_beta_prior, [45](#)
rllh_gamma_prior, [45](#)
rllh_normal_prior, [45](#)
rllh_pareto_prior, [45](#)
rng_manager, [47](#)
RngSeedT, [27](#)
sample_prior_stack, [45](#)
sgn, [46](#)
simulate_image_stack, [46](#)
square, [46](#)
StatsT, [27](#)
StencilVecT, [27](#)
StencilT, [27](#)
StringVecT, [27](#)
TERM_BLACK, [47](#)
TERM_BLUE, [47](#)
TERM_CYAN, [47](#)
TERM_DIM_BLACK, [47](#)
TERM_DIM_BLUE, [47](#)
TERM_DIM_CYAN, [47](#)
TERM_DIM_GREEN, [47](#)
TERM_DIM_MAGENTA, [48](#)
TERM_DIM_RED, [48](#)
TERM_DIM_WHITE, [48](#)
TERM_DIM_YELLOW, [48](#)
TERM_GREEN, [48](#)
TERM_MAGENTA, [48](#)
TERM_RED, [48](#)
TERM_WHITE, [48](#)
TERM_YELLOW, [49](#)
UniformDistT, [27](#)
VecFieldT, [27](#)
VecT, [27](#)
mappel::ArrayShapeError, [85](#)
 ArrayShapeError, [86](#)
mappel::ArraySizeError, [86](#)
 ArraySizeError, [87](#)
mappel::CGaussHeuristicEstimator
 CGaussHeuristicEstimator, [89](#)
 clear_stats, [89](#)
 compute_estimate, [89](#)
 compute_estimate_debug, [90](#)
 compute_profile_estimate, [90](#)
 Error, [89](#)
 estimate_max, [90](#), [91](#)
 estimate_max_debug, [91](#)
 estimate_max_stack, [91](#), [92](#)
 estimate_profile_stack, [92](#)
 exit_counts, [94](#)
 ExitCode, [89](#)
 FunctionChange, [89](#)
 get_debug_stats, [92](#)
 get_model, [92](#)
 get_stats, [92](#)
 GradRatio, [89](#)
 max_threads, [94](#)
 MaxBacktracks, [89](#)
 MaxIter, [89](#)
 model, [94](#)

- mtx, 94
- name, 93
- num_estimations, 94
- num_threads, 94
- NumExitCodes, 95
- record_exit_code, 93
- record_walltime, 93
- set_model, 93
- StepSize, 89
- Success, 89
- total_walltime, 95
- TrustRegionRadius, 89
- Unassigned, 89
- mappel::CGaussHeuristicEstimator< Model >, 87
- mappel::CGaussMLE< Model >, 95
- mappel::CGaussMLE
 - CGaussMLE, 97
 - clear_stats, 98
 - compute_estimate, 98
 - compute_estimate_debug, 98
 - compute_profile_estimate, 98
 - Error, 97
 - estimate_max, 99
 - estimate_max_debug, 100
 - estimate_max_stack, 100
 - estimate_profile_stack, 100
 - exit_counts, 102
 - ExitCode, 97
 - FunctionChange, 97
 - get_debug_stats, 100
 - get_model, 101
 - get_stats, 101
 - GradRatio, 97
 - max_threads, 102
 - MaxBacktracks, 97
 - MaxIter, 97
 - model, 102
 - mtx, 102
 - name, 101
 - num_estimations, 103
 - num_iterations, 103
 - num_threads, 103
 - NumExitCodes, 103
 - record_exit_code, 101
 - record_walltime, 101
 - set_model, 102
 - StepSize, 97
 - Success, 97
 - total_walltime, 103
 - TrustRegionRadius, 97
 - Unassigned, 97
- mappel::Estimator
 - ~Estimator, 106
 - clear_stats, 106
 - compute_estimate, 106
 - compute_estimate_debug, 107
 - compute_profile_estimate, 107
 - Error, 105
 - estimate_max, 107, 108
 - estimate_max_debug, 108
 - estimate_max_stack, 108, 109
 - estimate_profile_stack, 109
 - Estimator, 106
 - exit_counts, 110
 - ExitCode, 105
 - FunctionChange, 105
 - get_debug_stats, 109
 - get_model, 109
 - get_stats, 109
 - GradRatio, 105
 - MaxBacktracks, 105
 - MaxIter, 105
 - model, 110
 - name, 109
 - num_estimations, 111
 - NumExitCodes, 111
 - operator<<, 110
 - record_exit_code, 110
 - record_walltime, 110
 - set_model, 110
 - StepSize, 105
 - Success, 105
 - total_walltime, 111
 - TrustRegionRadius, 105
 - Unassigned, 105
- mappel::Estimator< Model >, 104
- mappel::Gauss1DMAP, 111
 - bound_theta, 118
 - bounded_theta, 118
 - bounded_theta_stack, 118
 - bounds_epsilon, 133
 - check_image_shape, 119
 - check_param_shape, 119
 - check_psf_sigma, 119
 - check_size, 120
 - default_alpha_sigma, 133
 - default_beta_pos, 133
 - default_intensity_kappa, 133
 - default_max_l, 133
 - default_mean_l, 134
 - default_pixel_mean_bg, 134
 - default_sigma_pos, 134
 - DefaultPriorType, 134
 - DefaultSeperableInitEstimator, 134
 - estimator_names, 134
 - eta_bg, 135
 - eta_l, 135
 - eta_x, 135

Gauss1DMap, 118
get_hyperparam_index, 120
get_hyperparam_names, 120
get_hyperparam_value, 120
get_hyperparams, 120
get_image_from_stack, 120
get_lbound, 120
get_mcmc_num_phases, 121
get_mcmc_sigma_scale, 121
get_num_hyperparams, 121
get_num_params, 121
get_num_pixels, 121
get_param_names, 121
get_prior, 121, 122
get_psf_sigma, 122
get_rng_generator, 122
get_rng_manager, 122
get_size, 122
get_size_image_stack, 123
get_stats, 123
get_ubound, 123
global_default_mcmc_sigma_scale, 135
global_max_mcmc_sigma_scale, 135
global_max_psf_sigma, 135
global_max_size, 136
global_min_psf_sigma, 136
global_min_size, 136
has_hyperparam, 123
ImageCoordT, 116
ImagePixelT, 116
ImageShapeT, 116
ImageSizeShapeT, 116
ImageSizeVecShapeT, 116
ImageSizeVecT, 116
ImageSizeT, 116
ImageStackShapeT, 116
ImageStackT, 117
ImageT, 117
initial_theta_estimate, 123
lbound, 136
make_default_prior, 123
make_default_prior_beta_position, 124
make_default_prior_normal_position, 124
make_image, 124
make_image_stack, 124
make_param, 124
make_param_mat, 125
make_param_mat_stack, 125
make_param_stack, 125
make_prior_beta_position, 125
make_prior_component_intensity, 126
make_prior_component_position_beta, 126
make_prior_component_position_normal, 126
make_prior_component_sigma, 126

make_prior_normal_position, 127
make_stencil, 127
ModelDataStackT, 117
ModelDataT, 117
name, 136
num_dim, 136
num_hyperparams, 137
num_params, 137
num_phases, 137
operator=, 127, 128
ParamVecT, 117
ParamT, 117
pixel_grad, 128
pixel_grad2, 128
pixel_hess, 128
pixel_hess_update, 128
pixel_model_value, 128
prior, 137
prior_types, 137
psf_sigma, 138
reflected_theta, 129
reflected_theta_stack, 129
rename_hyperparam, 129
sample_mcmc_candidate, 129
sample_prior, 129
set_background_mcmc_sampling, 130
set_bounds, 130
set_hyperparam_names, 130
set_hyperparam_value, 130
set_hyperparams, 130
set_image_in_stack, 130
set_intensity_mcmc_sampling, 131
set_lbound, 131
set_mcmc_num_phases, 131
set_mcmc_sigma_scale, 131
set_param_names, 131
set_prior, 131
set_psf_sigma, 132
set_rng_seed, 132
set_size, 132
set_ubound, 132
sigma_scale, 138
size, 138
StencilVecT, 117
theta_in_bounds, 132
theta_stack_in_bounds, 133
ubound, 138
mappel::Gauss1DMLE, 139
bound_theta, 146
bounded_theta, 146
bounded_theta_stack, 146
bounds_epsilon, 161
check_image_shape, 146
check_param_shape, 147

check_psf_sigma, 147
check_size, 147
default_alpha_sigma, 161
default_beta_pos, 161
default_intensity_kappa, 161
default_max_l, 161
default_mean_l, 162
default_pixel_mean_bg, 162
default_sigma_pos, 162
DefaultPriorType, 162
DefaultSeperableInitEstimator, 162
estimator_names, 162
eta_bg, 163
eta_l, 163
eta_x, 163
Gauss1DMLE, 145, 146
get_hyperparam_index, 147
get_hyperparam_names, 148
get_hyperparam_value, 148
get_hyperparams, 148
get_image_from_stack, 148
get_lbound, 148
get_mcmc_num_phases, 148
get_mcmc_sigma_scale, 148
get_num_hyperparams, 149
get_num_params, 149
get_num_pixels, 149
get_param_names, 149
get_prior, 149
get_psf_sigma, 149, 150
get_rng_generator, 150
get_rng_manager, 150
get_size, 150
get_size_image_stack, 150
get_stats, 150
get_ubound, 150
global_default_mcmc_sigma_scale, 163
global_max_mcmc_sigma_scale, 163
global_max_psf_sigma, 163
global_max_size, 164
global_min_psf_sigma, 164
global_min_size, 164
has_hyperparam, 151
ImageCoordT, 143
ImagePixelT, 143
ImageShapeT, 143
ImageSizeShapeT, 143
ImageSizeVecShapeT, 144
ImageSizeVecT, 144
ImageSizeT, 144
ImageStackShapeT, 144
ImageStackT, 144
ImageT, 144
initial_theta_estimate, 151
lbound, 164
make_default_prior, 151
make_default_prior_beta_position, 151
make_default_prior_normal_position, 151
make_image, 152
make_image_stack, 152
make_param, 152
make_param_mat, 152
make_param_mat_stack, 153
make_param_stack, 153
make_prior_beta_position, 153
make_prior_component_intensity, 153
make_prior_component_position_beta, 154
make_prior_component_position_normal, 154
make_prior_component_sigma, 154
make_prior_normal_position, 154
make_stencil, 155
ModelDataStackT, 144
ModelDataT, 145
name, 164
num_dim, 164
num_hyperparams, 165
num_params, 165
num_phases, 165
operator=, 155
ParamVecT, 145
ParamT, 145
pixel_grad, 156
pixel_grad2, 156
pixel_hess, 156
pixel_hess_update, 156
pixel_model_value, 156
prior, 165
prior_types, 165
psf_sigma, 166
reflected_theta, 156
reflected_theta_stack, 157
rename_hyperparam, 157
sample_mcmc_candidate, 157
sample_prior, 157
set_background_mcmc_sampling, 157
set_bounds, 158
set_hyperparam_names, 158
set_hyperparam_value, 158
set_hyperparams, 158
set_image_in_stack, 158
set_intensity_mcmc_sampling, 158
set_lbound, 159
set_mcmc_num_phases, 159
set_mcmc_sigma_scale, 159
set_param_names, 159
set_prior, 159
set_psf_sigma, 160
set_rng_seed, 160

- set_size, 160
- set_ubound, 160
- sigma_scale, 166
- size, 166
- StencilVecT, 145
- theta_in_bounds, 160
- theta_stack_in_bounds, 161
- ubound, 166
- mappel::Gauss1DModel, 167
 - bound_theta, 173
 - bounded_theta, 173
 - bounded_theta_stack, 174
 - bounds_epsilon, 188
 - check_image_shape, 174
 - check_param_shape, 174
 - check_psf_sigma, 174, 175
 - check_size, 175
 - default_alpha_sigma, 188
 - default_beta_pos, 188
 - default_intensity_kappa, 188
 - default_max_l, 188
 - default_mean_l, 189
 - default_pixel_mean_bg, 189
 - default_sigma_pos, 189
 - DefaultPriorType, 189
 - DefaultSeperableInitEstimator, 189
 - eta_bg, 189
 - eta_l, 190
 - eta_x, 190
 - Gauss1DModel, 173
 - get_hyperparam_index, 175
 - get_hyperparam_names, 175
 - get_hyperparam_value, 175
 - get_hyperparams, 175
 - get_image_from_stack, 176
 - get_lbound, 176
 - get_mcmc_num_phases, 176
 - get_mcmc_sigma_scale, 176
 - get_num_hyperparams, 176
 - get_num_params, 176
 - get_num_pixels, 176
 - get_param_names, 177
 - get_prior, 177
 - get_psf_sigma, 177
 - get_rng_generator, 177
 - get_rng_manager, 177
 - get_size, 178
 - get_size_image_stack, 178
 - get_stats, 178
 - get_ubound, 178
 - global_default_mcmc_sigma_scale, 190
 - global_max_mcmc_sigma_scale, 190
 - global_max_psf_sigma, 190
 - global_max_size, 191
 - global_min_psf_sigma, 191
 - global_min_size, 191
 - has_hyperparam, 178
 - ImageCoordT, 171
 - ImagePixelT, 171
 - ImageShapeT, 171
 - ImageSizeShapeT, 172
 - ImageSizeVecShapeT, 172
 - ImageSizeVecT, 172
 - ImageSizeT, 172
 - ImageStackShapeT, 172
 - ImageStackT, 172
 - ImageT, 172
 - initial_theta_estimate, 178, 179
 - lbound, 191
 - make_default_prior, 179
 - make_default_prior_beta_position, 179
 - make_default_prior_normal_position, 179
 - make_image, 179
 - make_image_stack, 179
 - make_param, 179, 180
 - make_param_mat, 180
 - make_param_mat_stack, 180
 - make_param_stack, 180
 - make_prior_beta_position, 181
 - make_prior_component_intensity, 181
 - make_prior_component_position_beta, 181
 - make_prior_component_position_normal, 181
 - make_prior_component_sigma, 181
 - make_prior_normal_position, 182
 - make_stencil, 182
 - num_dim, 191
 - num_hyperparams, 192
 - num_params, 192
 - num_phases, 192
 - operator=, 182, 183
 - ParamVecT, 173
 - ParamT, 173
 - pixel_grad, 183
 - pixel_grad2, 183
 - pixel_hess, 183
 - pixel_hess_update, 183
 - pixel_model_value, 183
 - prior, 192
 - prior_types, 192
 - psf_sigma, 193
 - reflected_theta, 183
 - reflected_theta_stack, 184
 - rename_hyperparam, 184
 - sample_mcmc_candidate, 184
 - sample_prior, 184
 - set_background_mcmc_sampling, 184
 - set_bounds, 185
 - set_hyperparam_names, 185

- set_hyperparam_value, 185
- set_hyperparams, 185
- set_image_in_stack, 185
- set_intensity_mcmc_sampling, 185
- set_lbound, 186
- set_mcmc_num_phases, 186
- set_mcmc_sigma_scale, 186
- set_param_names, 186
- set_prior, 186
- set_psf_sigma, 187
- set_rng_seed, 187
- set_size, 187
- set_ubound, 187
- sigma_scale, 193
- size, 193
- StencilVecT, 173
- theta_in_bounds, 187
- theta_stack_in_bounds, 188
- ubound, 193
- mappel::Gauss1DModel::Stencil, 746
 - bg, 747
 - compute_derivatives, 747
 - DXS, 748
 - derivatives_computed, 748
 - DX, 748
 - dx, 748
 - Gx, 748
 - I, 747
 - model, 749
 - operator<<, 748
 - ParamT, 746
 - Stencil, 747
 - theta, 749
 - X, 749
 - x, 747
- mappel::Gauss1DsMAP, 194
 - bound_theta, 201
 - bounded_theta, 201
 - bounded_theta_stack, 201
 - bounds_epsilon, 216
 - check_image_shape, 201
 - check_param_shape, 202
 - check_psf_sigma, 202
 - check_size, 202
 - default_alpha_sigma, 216
 - default_beta_pos, 216
 - default_intensity_kappa, 217
 - default_max_I, 217
 - default_mean_I, 217
 - default_pixel_mean_bg, 217
 - default_sigma_pos, 217
 - DefaultPriorType, 217
 - DefaultSeperableInitEstimator, 218
 - estimator_names, 218
 - eta_bg, 218
 - eta_I, 218
 - eta_sigma, 218
 - eta_x, 218
 - Gauss1DsMAP, 200, 201
 - get_hyperparam_index, 202
 - get_hyperparam_names, 203
 - get_hyperparam_value, 203
 - get_hyperparams, 203
 - get_image_from_stack, 203
 - get_lbound, 203
 - get_max_sigma, 203
 - get_mcmc_num_phases, 203
 - get_mcmc_sigma_scale, 204
 - get_min_sigma, 204
 - get_num_hyperparams, 204
 - get_num_params, 204
 - get_num_pixels, 204
 - get_param_names, 204
 - get_prior, 204, 205
 - get_rng_generator, 205
 - get_rng_manager, 205
 - get_size, 205
 - get_size_image_stack, 205
 - get_stats, 205
 - get_ubound, 206
 - global_default_mcmc_sigma_scale, 219
 - global_max_mcmc_sigma_scale, 219
 - global_max_psf_sigma, 219
 - global_max_size, 219
 - global_min_psf_sigma, 219
 - global_min_size, 219
 - has_hyperparam, 206
 - ImageCoordT, 198
 - ImagePixelT, 198
 - ImageShapeT, 198
 - ImageSizeShapeT, 198
 - ImageSizeVecShapeT, 199
 - ImageSizeVecT, 199
 - ImageSizeT, 199
 - ImageStackShapeT, 199
 - ImageStackT, 199
 - ImageT, 199
 - initial_theta_estimate, 206
 - lbound, 220
 - make_default_prior, 206
 - make_default_prior_beta_position, 206
 - make_default_prior_normal_position, 207
 - make_image, 207
 - make_image_stack, 207
 - make_param, 207
 - make_param_mat, 207, 208
 - make_param_mat_stack, 208
 - make_param_stack, 208

make_prior_beta_position, 208
make_prior_component_intensity, 208
make_prior_component_position_beta, 209
make_prior_component_position_normal, 209
make_prior_component_sigma, 209
make_prior_normal_position, 209
make_stencil, 210
ModelDataStackT, 199
ModelDataT, 200
name, 220
num_dim, 220
num_hyperparams, 220
num_params, 220
num_phases, 220
operator=, 210
ParamVecT, 200
ParamT, 200
pixel_grad, 211
pixel_grad2, 211
pixel_hess, 211
pixel_hess_update, 211
pixel_model_value, 211
prior, 221
prior_types, 221
reflected_theta, 211
reflected_theta_stack, 212
rename_hyperparam, 212
sample_mcmc_candidate, 212
sample_prior, 212
set_background_mcmc_sampling, 212
set_bounds, 213
set_hyperparam_names, 213
set_hyperparam_value, 213
set_hyperparams, 213
set_image_in_stack, 213
set_intensity_mcmc_sampling, 213
set_lbound, 214
set_max_sigma, 214
set_mcmc_num_phases, 214
set_mcmc_sigma_scale, 214
set_min_sigma, 214, 215
set_param_names, 215
set_prior, 215
set_rng_seed, 215
set_size, 215
set_ubound, 216
sigma_scale, 221
size, 221
StencilVecT, 200
theta_in_bounds, 216
theta_stack_in_bounds, 216
ubound, 222
mappel::Gauss1DsMLE, 222
 bound_theta, 229
 bounded_theta, 229
 bounded_theta_stack, 229
 bounds_epsilon, 244
 check_image_shape, 230
 check_param_shape, 230
 check_psf_sigma, 230
 check_size, 231
 default_alpha_sigma, 244
 default_beta_pos, 245
 default_intensity_kappa, 245
 default_max_l, 245
 default_mean_l, 245
 default_pixel_mean_bg, 245
 default_sigma_pos, 245
 DefaultPriorType, 246
 DefaultSeperableInitEstimator, 246
 estimator_names, 246
 eta_bg, 246
 eta_l, 246
 eta_sigma, 246
 eta_x, 247
 Gauss1DsMLE, 229
 get_hyperparam_index, 231
 get_hyperparam_names, 231
 get_hyperparam_value, 231
 get_hyperparams, 231
 get_image_from_stack, 231
 get_lbound, 231
 get_max_sigma, 232
 get_mcmc_num_phases, 232
 get_mcmc_sigma_scale, 232
 get_min_sigma, 232
 get_num_hyperparams, 232
 get_num_params, 232
 get_num_pixels, 232
 get_param_names, 233
 get_prior, 233
 get_rng_generator, 233
 get_rng_manager, 233
 get_size, 233
 get_size_image_stack, 234
 get_stats, 234
 get_ubound, 234
 global_default_mcmc_sigma_scale, 247
 global_max_mcmc_sigma_scale, 247
 global_max_psf_sigma, 247
 global_max_size, 247
 global_min_psf_sigma, 247
 global_min_size, 248
 has_hyperparam, 234
 ImageCoordT, 227
 ImagePixelT, 227
 ImageShapeT, 227
 ImageSizeShapeT, 227

ImageSizeVecShapeT, [227](#)
 ImageSizeVecT, [227](#)
 ImageSizeT, [227](#)
 ImageStackShapeT, [227](#)
 ImageStackT, [228](#)
 ImageT, [228](#)
 initial_theta_estimate, [234](#)
 lbound, [248](#)
 make_default_prior, [234](#)
 make_default_prior_beta_position, [235](#)
 make_default_prior_normal_position, [235](#)
 make_image, [235](#)
 make_image_stack, [235](#)
 make_param, [235](#)
 make_param_mat, [236](#)
 make_param_mat_stack, [236](#)
 make_param_stack, [236](#)
 make_prior_beta_position, [236](#)
 make_prior_component_intensity, [237](#)
 make_prior_component_position_beta, [237](#)
 make_prior_component_position_normal, [237](#)
 make_prior_component_sigma, [237](#)
 make_prior_normal_position, [238](#)
 make_stencil, [238](#)
 ModelDataStackT, [228](#)
 ModelDataT, [228](#)
 name, [248](#)
 num_dim, [248](#)
 num_hyperparams, [248](#)
 num_params, [248](#)
 num_phases, [249](#)
 operator=, [238](#), [239](#)
 ParamVecT, [228](#)
 ParamT, [228](#)
 pixel_grad, [239](#)
 pixel_grad2, [239](#)
 pixel_hess, [239](#)
 pixel_hess_update, [239](#)
 pixel_model_value, [239](#)
 prior, [249](#)
 prior_types, [249](#)
 reflected_theta, [240](#)
 reflected_theta_stack, [240](#)
 rename_hyperparam, [240](#)
 sample_mcmc_candidate, [240](#)
 sample_prior, [240](#)
 set_background_mcmc_sampling, [241](#)
 set_bounds, [241](#)
 set_hyperparam_names, [241](#)
 set_hyperparam_value, [241](#)
 set_hyperparams, [241](#)
 set_image_in_stack, [241](#)
 set_intensity_mcmc_sampling, [242](#)
 set_lbound, [242](#)
 set_max_sigma, [242](#)
 set_mcmc_num_phases, [242](#)
 set_mcmc_sigma_scale, [242](#)
 set_min_sigma, [243](#)
 set_param_names, [243](#)
 set_prior, [243](#)
 set_rng_seed, [243](#)
 set_size, [243](#), [244](#)
 set_ubound, [244](#)
 sigma_scale, [249](#)
 size, [250](#)
 StencilVecT, [228](#)
 theta_in_bounds, [244](#)
 theta_stack_in_bounds, [244](#)
 ubound, [250](#)
 mappel::Gauss1DsModel, [250](#)
 bound_theta, [257](#)
 bounded_theta, [257](#)
 bounded_theta_stack, [257](#)
 bounds_epsilon, [272](#)
 check_image_shape, [257](#)
 check_param_shape, [257](#)
 check_psf_sigma, [258](#)
 check_size, [258](#)
 default_alpha_sigma, [272](#)
 default_beta_pos, [272](#)
 default_intensity_kappa, [272](#)
 default_max_l, [272](#)
 default_mean_l, [272](#)
 default_pixel_mean_bg, [272](#)
 default_sigma_pos, [273](#)
 DefaultPriorType, [273](#)
 DefaultSeperableInitEstimator, [273](#)
 eta_bg, [273](#)
 eta_l, [273](#)
 eta_sigma, [273](#)
 eta_x, [274](#)
 Gauss1DsModel, [256](#)
 get_hyperparam_index, [258](#)
 get_hyperparam_names, [258](#)
 get_hyperparam_value, [258](#)
 get_hyperparams, [259](#)
 get_image_from_stack, [259](#)
 get_lbound, [259](#)
 get_max_sigma, [259](#)
 get_mcmc_num_phases, [259](#)
 get_mcmc_sigma_scale, [259](#)
 get_min_sigma, [259](#)
 get_num_hyperparams, [260](#)
 get_num_params, [260](#)
 get_num_pixels, [260](#)
 get_param_names, [260](#)
 get_prior, [260](#)
 get_rng_generator, [260](#)

- get_rng_manager, [261](#)
- get_size, [261](#)
- get_size_image_stack, [261](#)
- get_stats, [261](#)
- get_ubound, [261](#)
- global_default_mcmc_sigma_scale, [274](#)
- global_max_mcmc_sigma_scale, [274](#)
- global_max_psf_sigma, [274](#)
- global_max_size, [274](#)
- global_min_psf_sigma, [274](#)
- global_min_size, [275](#)
- has_hyperparam, [261](#)
- ImageCoordT, [254](#)
- ImagePixelT, [254](#)
- ImageShapeT, [255](#)
- ImageSizeShapeT, [255](#)
- ImageSizeVecShapeT, [255](#)
- ImageSizeVecT, [255](#)
- ImageSizeT, [255](#)
- ImageStackShapeT, [255](#)
- ImageStackT, [255](#)
- ImageT, [256](#)
- initial_theta_estimate, [262](#)
- lbound, [275](#)
- make_default_prior, [262](#)
- make_default_prior_beta_position, [262](#)
- make_default_prior_normal_position, [262](#)
- make_image, [262](#)
- make_image_stack, [263](#)
- make_param, [263](#)
- make_param_mat, [263](#)
- make_param_mat_stack, [263](#)
- make_param_stack, [264](#)
- make_prior_beta_position, [264](#)
- make_prior_component_intensity, [264](#)
- make_prior_component_position_beta, [264](#)
- make_prior_component_position_normal, [265](#)
- make_prior_component_sigma, [265](#)
- make_prior_normal_position, [265](#)
- make_stencil, [265](#)
- num_dim, [275](#)
- num_hyperparams, [275](#)
- num_params, [275](#)
- num_phases, [275](#)
- operator=, [266](#)
- ParamVecT, [256](#)
- ParamT, [256](#)
- pixel_grad, [266](#)
- pixel_grad2, [266](#)
- pixel_hess, [266](#)
- pixel_hess_update, [267](#)
- pixel_model_value, [267](#)
- prior, [276](#)
- prior_types, [276](#)

- reflected_theta, [267](#)
- reflected_theta_stack, [267](#)
- rename_hyperparam, [267](#)
- sample_mcmc_candidate, [267](#)
- sample_prior, [268](#)
- set_background_mcmc_sampling, [268](#)
- set_bounds, [268](#)
- set_hyperparam_names, [268](#)
- set_hyperparam_value, [268](#)
- set_hyperparams, [269](#)
- set_image_in_stack, [269](#)
- set_intensity_mcmc_sampling, [269](#)
- set_lbound, [269](#)
- set_max_sigma, [269](#)
- set_mcmc_num_phases, [270](#)
- set_mcmc_sigma_scale, [270](#)
- set_min_sigma, [270](#)
- set_param_names, [270](#)
- set_prior, [270](#)
- set_rng_seed, [271](#)
- set_size, [271](#)
- set_ubound, [271](#)
- sigma_scale, [276](#)
- size, [276](#)
- StencilVecT, [256](#)
- theta_in_bounds, [271](#)
- theta_stack_in_bounds, [271](#)
- ubound, [277](#)
- mappel::Gauss1DsModel::Stencil, [749](#)
 - bg, [751](#)
 - compute_derivatives, [751](#)
 - DXS2, [752](#)
 - DXSX, [752](#)
 - DXS, [752](#)
 - derivatives_computed, [752](#)
 - DX, [752](#)
 - dx, [752](#)
 - Gx, [752](#)
 - I, [751](#)
 - model, [753](#)
 - operator<<, [752](#)
 - ParamT, [750](#)
 - sigma, [751](#)
 - Stencil, [750](#)
 - theta, [753](#)
 - X, [753](#)
 - x, [751](#)
- mappel::Gauss2DMAP, [277](#)
 - bound_theta, [285](#)
 - bounded_theta, [285](#)
 - bounded_theta_stack, [285](#)
 - bounds_epsilon, [300](#)
 - check_image_shape, [285](#)
 - check_param_shape, [285](#), [286](#)

check_psf_sigma, 286
check_size, 286
debug_internal_sum_model_x, 286
debug_internal_sum_model_y, 286
default_alpha_sigma, 300
default_beta_pos, 300
default_intensity_kappa, 301
default_max_l, 301
default_mean_l, 301
default_pixel_mean_bg, 301
default_sigma_pos, 301
DefaultPriorType, 301
DefaultSeperableInitEstimator, 302
estimator_names, 302
eta_bg, 302
eta_l, 302
eta_x, 302
eta_y, 302
Gauss1DSumModelT, 282
Gauss2DMAP, 284, 285
get_hyperparam_index, 287
get_hyperparam_names, 287
get_hyperparam_value, 287
get_hyperparams, 287
get_image_from_stack, 287
get_lbound, 287
get_mcmc_num_phases, 287
get_mcmc_sigma_scale, 288
get_num_hyperparams, 288
get_num_params, 288
get_num_pixels, 288
get_param_names, 288
get_prior, 288
get_psf_sigma, 289
get_rng_generator, 289
get_rng_manager, 289
get_size, 289
get_size_image_stack, 289
get_stats, 290
get_ubound, 290
global_default_mcmc_sigma_scale, 303
global_max_mcmc_sigma_scale, 303
global_max_psf_sigma, 303
global_max_size, 303
global_min_psf_sigma, 303
global_min_size, 303
has_hyperparam, 290
ImageCoordT, 282
ImagePixelT, 282
ImageShapeT, 282
ImageSizeShapeT, 282
ImageSizeVecShapeT, 282
ImageSizeVecT, 283
ImageSizeT, 282
ImageStackShapeT, 283
ImageStackT, 283
ImageT, 283
initial_theta_estimate, 290
lbound, 304
make_default_prior, 291
make_default_prior_beta_position, 291
make_default_prior_normal_position, 291
make_image, 291
make_image_stack, 291
make_internal_1Dsum_estimator, 291
make_param, 292
make_param_mat, 292
make_param_mat_stack, 292
make_param_stack, 293
make_prior_beta_position, 293
make_prior_component_intensity, 293
make_prior_component_position_beta, 293
make_prior_component_position_normal, 294
make_prior_component_sigma, 294
make_prior_normal_position, 294
make_stencil, 294
ModelDataStackT, 283
ModelDataT, 283
name, 304
num_dim, 304
num_hyperparams, 304
num_params, 304
num_phases, 304
operator=, 295
ParamVecT, 284
ParamT, 283
pixel_grad, 295
pixel_grad2, 295
pixel_hess, 295
pixel_hess_update, 296
pixel_model_value, 296
prior, 305
prior_types, 305
psf_sigma, 305
reflected_theta, 296
reflected_theta_stack, 296
rename_hyperparam, 296
sample_mcmc_candidate, 296
sample_prior, 297
set_background_mcmc_sampling, 297
set_bounds, 297
set_hyperparam_names, 297
set_hyperparam_value, 297
set_hyperparams, 298
set_image_in_stack, 298
set_intensity_mcmc_sampling, 298
set_lbound, 298
set_mcmc_num_phases, 298

- set_mcmc_sigma_scale, 298
- set_param_names, 298
- set_prior, 299
- set_psf_sigma, 299
- set_rng_seed, 299
- set_size, 299
- set_ubound, 299
- sigma_scale, 305
- size, 306
- StencilVecT, 284
- theta_in_bounds, 299
- theta_stack_in_bounds, 300
- ubound, 306
- update_internal_1Dsum_estimators, 300
- x_model, 306
- y_model, 307
- mappel::Gauss2DMLE, 307
 - bound_theta, 315
 - bounded_theta, 315
 - bounded_theta_stack, 315
 - bounds_epsilon, 330
 - check_image_shape, 315
 - check_param_shape, 315, 316
 - check_psf_sigma, 316
 - check_size, 316
 - debug_internal_sum_model_x, 316
 - debug_internal_sum_model_y, 316
 - default_alpha_sigma, 330
 - default_beta_pos, 330
 - default_intensity_kappa, 331
 - default_max_l, 331
 - default_mean_l, 331
 - default_pixel_mean_bg, 331
 - default_sigma_pos, 331
 - DefaultPriorType, 331
 - DefaultSeperableInitEstimator, 332
 - estimator_names, 332
 - eta_bg, 332
 - eta_l, 332
 - eta_x, 332
 - eta_y, 332
 - Gauss1DSumModelT, 312
 - Gauss2DMLE, 314
 - get_hyperparam_index, 317
 - get_hyperparam_names, 317
 - get_hyperparam_value, 317
 - get_hyperparams, 317
 - get_image_from_stack, 317
 - get_lbound, 317
 - get_mcmc_num_phases, 317
 - get_mcmc_sigma_scale, 318
 - get_num_hyperparams, 318
 - get_num_params, 318
 - get_num_pixels, 318
 - get_param_names, 318
 - get_prior, 318
 - get_psf_sigma, 319
 - get_rng_generator, 319
 - get_rng_manager, 319
 - get_size, 319
 - get_size_image_stack, 319
 - get_stats, 320
 - get_ubound, 320
 - global_default_mcmc_sigma_scale, 333
 - global_max_mcmc_sigma_scale, 333
 - global_max_psf_sigma, 333
 - global_max_size, 333
 - global_min_psf_sigma, 333
 - global_min_size, 333
 - has_hyperparam, 320
 - ImageCoordT, 312
 - ImagePixelT, 312
 - ImageShapeT, 312
 - ImageSizeShapeT, 312
 - ImageSizeVecShapeT, 312
 - ImageSizeVecT, 313
 - ImageSizeT, 312
 - ImageStackShapeT, 313
 - ImageStackT, 313
 - ImageT, 313
 - initial_theta_estimate, 320
 - lbound, 334
 - make_default_prior, 321
 - make_default_prior_beta_position, 321
 - make_default_prior_normal_position, 321
 - make_image, 321
 - make_image_stack, 321
 - make_internal_1Dsum_estimator, 321
 - make_param, 322
 - make_param_mat, 322
 - make_param_mat_stack, 322
 - make_param_stack, 323
 - make_prior_beta_position, 323
 - make_prior_component_intensity, 323
 - make_prior_component_position_beta, 323
 - make_prior_component_position_normal, 324
 - make_prior_component_sigma, 324
 - make_prior_normal_position, 324
 - make_stencil, 324
 - ModelDataStackT, 313
 - ModelDataT, 313
 - name, 334
 - num_dim, 334
 - num_hyperparams, 334
 - num_params, 334
 - num_phases, 334
 - operator=, 325
 - ParamVecT, 314

- ParamT, 313
- pixel_grad, 325
- pixel_grad2, 325
- pixel_hess, 325
- pixel_hess_update, 326
- pixel_model_value, 326
- prior, 335
- prior_types, 335
- psf_sigma, 335
- reflected_theta, 326
- reflected_theta_stack, 326
- rename_hyperparam, 326
- sample_mcmc_candidate, 326
- sample_prior, 327
- set_background_mcmc_sampling, 327
- set_bounds, 327
- set_hyperparam_names, 327
- set_hyperparam_value, 327
- set_hyperparams, 328
- set_image_in_stack, 328
- set_intensity_mcmc_sampling, 328
- set_lbound, 328
- set_mcmc_num_phases, 328
- set_mcmc_sigma_scale, 328
- set_param_names, 328
- set_prior, 329
- set_psf_sigma, 329
- set_rng_seed, 329
- set_size, 329
- set_ubound, 329
- sigma_scale, 335
- size, 336
- StencilVecT, 314
- theta_in_bounds, 329
- theta_stack_in_bounds, 330
- ubound, 336
- update_internal_1Dsum_estimators, 330
- x_model, 336
- y_model, 337
- mappel::Gauss2DModel, 337
 - bound_theta, 344
 - bounded_theta, 344
 - bounded_theta_stack, 344
 - bounds_epsilon, 359
 - check_image_shape, 344
 - check_param_shape, 345
 - check_psf_sigma, 345
 - check_size, 345
 - debug_internal_sum_model_x, 345
 - debug_internal_sum_model_y, 346
 - default_alpha_sigma, 359
 - default_beta_pos, 359
 - default_intensity_kappa, 359
 - default_max_l, 360
 - default_mean_l, 360
 - default_pixel_mean_bg, 360
 - default_sigma_pos, 360
 - DefaultPriorType, 360
 - DefaultSeperableInitEstimator, 360
 - eta_bg, 361
 - eta_l, 361
 - eta_x, 361
 - eta_y, 361
 - Gauss1DSumModelT, 341
 - Gauss2DModel, 343, 344
 - get_hyperparam_index, 346
 - get_hyperparam_names, 346
 - get_hyperparam_value, 346
 - get_hyperparams, 346
 - get_image_from_stack, 346
 - get_lbound, 346
 - get_mcmc_num_phases, 347
 - get_mcmc_sigma_scale, 347
 - get_num_hyperparams, 347
 - get_num_params, 347
 - get_num_pixels, 347
 - get_param_names, 347
 - get_prior, 347, 348
 - get_psf_sigma, 348
 - get_rng_generator, 348
 - get_rng_manager, 348
 - get_size, 348
 - get_size_image_stack, 349
 - get_stats, 349
 - get_ubound, 349
 - global_default_mcmc_sigma_scale, 361
 - global_max_mcmc_sigma_scale, 362
 - global_max_psf_sigma, 362
 - global_max_size, 362
 - global_min_psf_sigma, 362
 - global_min_size, 362
 - has_hyperparam, 349
 - ImageCoordT, 341
 - ImagePixelT, 342
 - ImageShapeT, 342
 - ImageSizeShapeT, 342
 - ImageSizeVecShapeT, 342
 - ImageSizeVecT, 342
 - ImageSizeT, 342
 - ImageStackShapeT, 342
 - ImageStackT, 343
 - ImageT, 343
 - initial_theta_estimate, 349, 350
 - lbound, 362
 - make_default_prior, 350
 - make_default_prior_beta_position, 350
 - make_default_prior_normal_position, 350
 - make_image, 350

- [make_image_stack](#), 350
- [make_internal_1Dsum_estimator](#), 351
- [make_param](#), 351
- [make_param_mat](#), 351
- [make_param_mat_stack](#), 351, 352
- [make_param_stack](#), 352
- [make_prior_beta_position](#), 352
- [make_prior_component_intensity](#), 352
- [make_prior_component_position_beta](#), 352
- [make_prior_component_position_normal](#), 353
- [make_prior_component_sigma](#), 353
- [make_prior_normal_position](#), 353
- [make_stencil](#), 353
- [num_dim](#), 363
- [num_hyperparams](#), 363
- [num_params](#), 363
- [num_phases](#), 363
- [operator=](#), 354
- [ParamVecT](#), 343
- [ParamT](#), 343
- [pixel_grad](#), 354
- [pixel_grad2](#), 354
- [pixel_hess](#), 354
- [pixel_hess_update](#), 355
- [pixel_model_value](#), 355
- [prior](#), 363
- [prior_types](#), 364
- [psf_sigma](#), 364
- [reflected_theta](#), 355
- [reflected_theta_stack](#), 355
- [rename_hyperparam](#), 355
- [sample_mcmc_candidate](#), 355
- [sample_prior](#), 356
- [set_background_mcmc_sampling](#), 356
- [set_bounds](#), 356
- [set_hyperparam_names](#), 356
- [set_hyperparam_value](#), 356
- [set_hyperparams](#), 357
- [set_image_in_stack](#), 357
- [set_intensity_mcmc_sampling](#), 357
- [set_lbound](#), 357
- [set_mcmc_num_phases](#), 357
- [set_mcmc_sigma_scale](#), 357
- [set_param_names](#), 357
- [set_prior](#), 358
- [set_psf_sigma](#), 358
- [set_rng_seed](#), 358
- [set_size](#), 358
- [set_ubound](#), 358
- [sigma_scale](#), 364
- [size](#), 364
- [StencilVecT](#), 343
- [theta_in_bounds](#), 358
- [theta_stack_in_bounds](#), 359
- [ubound](#), 365
- [update_internal_1Dsum_estimators](#), 359
- [x_model](#), 365
- [y_model](#), 365
- [mappel::Gauss2DModel::Stencil](#), 764
 - [bg](#), 766
 - [compute_derivatives](#), 766
 - [DXS](#), 767
 - [DYS](#), 768
 - [derivatives_computed](#), 767
 - [DX](#), 767
 - [dx](#), 767
 - [DY](#), 768
 - [dy](#), 767
 - [Gx](#), 768
 - [Gy](#), 768
 - [I](#), 766
 - [model](#), 768
 - [operator<<](#), 767
 - [ParamT](#), 765
 - [Stencil](#), 765
 - [theta](#), 768
 - [X](#), 768
 - [x](#), 766
 - [Y](#), 768
 - [y](#), 766
- [mappel::Gauss2DsMAP](#), 366
 - [bound_theta](#), 373
 - [bounded_theta](#), 373
 - [bounded_theta_stack](#), 374
 - [bounds_epsilon](#), 391
 - [check_image_shape](#), 374
 - [check_param_shape](#), 374
 - [check_psf_sigma](#), 374, 375
 - [check_size](#), 375
 - [compute_max_sigma_ratio](#), 375
 - [debug_internal_sum_model_x](#), 375
 - [debug_internal_sum_model_y](#), 375
 - [default_alpha_sigma](#), 391
 - [default_beta_pos](#), 391
 - [default_intensity_kappa](#), 391
 - [default_max_I](#), 391
 - [default_mean_I](#), 391
 - [default_pixel_mean_bg](#), 392
 - [default_sigma_pos](#), 392
 - [DefaultPriorType](#), 392
 - [DefaultSeperableInitEstimator](#), 392
 - [estimator_names](#), 392
 - [eta_bg](#), 392
 - [eta_I](#), 393
 - [eta_sigma](#), 393
 - [eta_x](#), 393
 - [eta_y](#), 393
 - [Gauss1DSumModelT](#), 370

Gauss2DsMAP, 373
get_hyperparam_index, 376
get_hyperparam_names, 376
get_hyperparam_value, 376
get_hyperparams, 376
get_image_from_stack, 376
get_lbound, 376
get_max_sigma, 376, 377
get_max_sigma_ratio, 377
get_mcmc_num_phases, 377
get_mcmc_sigma_scale, 377
get_min_sigma, 377
get_num_hyperparams, 377
get_num_params, 378
get_num_pixels, 378
get_param_names, 378
get_prior, 378
get_rng_generator, 378
get_rng_manager, 378
get_size, 379
get_size_image_stack, 379
get_stats, 379
get_ubound, 379
global_default_mcmc_sigma_scale, 393
global_max_mcmc_sigma_scale, 394
global_max_psf_sigma, 394
global_max_size, 394
global_min_psf_sigma, 394
global_min_size, 394
has_hyperparam, 379
ImageCoordT, 370
ImagePixelT, 371
ImageShapeT, 371
ImageSizeShapeT, 371
ImageSizeVecShapeT, 371
ImageSizeVecT, 371
ImageSizeT, 371
ImageStackShapeT, 371
ImageStackT, 372
ImageT, 372
initial_theta_estimate, 379, 380
lbound, 394
make_default_prior, 380
make_default_prior_beta_position, 380
make_default_prior_normal_position, 380
make_image, 381
make_image_stack, 381
make_internal_1Dsum_estimator, 381
make_param, 381
make_param_mat, 382
make_param_mat_stack, 382
make_param_stack, 382
make_prior_beta_position, 382
make_prior_component_intensity, 383
make_prior_component_position_beta, 383
make_prior_component_position_normal, 383
make_prior_component_sigma, 383
make_prior_normal_position, 384
make_stencil, 384
min_sigma, 395
ModelDataStackT, 372
ModelDataT, 372
name, 395
num_dim, 395
num_hyperparams, 395
num_params, 395
num_phases, 395
operator=, 384, 385
ParamVecT, 372
ParamT, 372
pixel_grad, 385
pixel_grad2, 385
pixel_hess, 385
pixel_hess_update, 385
pixel_model_value, 386
prior, 396
prior_types, 396
reflected_theta, 386
reflected_theta_stack, 386
rename_hyperparam, 386
sample_mcmc_candidate, 386
sample_prior, 387
set_background_mcmc_sampling, 387
set_bounds, 387
set_hyperparam_names, 387
set_hyperparam_value, 387
set_hyperparams, 388
set_image_in_stack, 388
set_intensity_mcmc_sampling, 388
set_lbound, 388
set_max_sigma, 388
set_max_sigma_ratio, 388
set_mcmc_num_phases, 389
set_mcmc_sigma_scale, 389
set_min_sigma, 389
set_param_names, 389
set_prior, 389
set_rng_seed, 390
set_size, 390
set_ubound, 390
sigma_scale, 396
size, 396
StencilVecT, 372
theta_in_bounds, 390
theta_stack_in_bounds, 390
ubound, 387
update_internal_1Dsum_estimators, 390
x_model, 397

y_model, 397
mappel::Gauss2DsMLE, 398
 bound_theta, 405
 bounded_theta, 405
 bounded_theta_stack, 406
 bounds_epsilon, 423
 check_image_shape, 406
 check_param_shape, 406
 check_psf_sigma, 406, 407
 check_size, 407
 compute_max_sigma_ratio, 407
 debug_internal_sum_model_x, 407
 debug_internal_sum_model_y, 407
 default_alpha_sigma, 423
 default_beta_pos, 423
 default_intensity_kappa, 423
 default_max_l, 423
 default_mean_l, 423
 default_pixel_mean_bg, 424
 default_sigma_pos, 424
 DefaultPriorType, 424
 DefaultSeperableInitEstimator, 424
 estimator_names, 424
 eta_bg, 424
 eta_l, 425
 eta_sigma, 425
 eta_x, 425
 eta_y, 425
 Gauss1DSumModelT, 402
 Gauss2DsMLE, 405
 get_hyperparam_index, 408
 get_hyperparam_names, 408
 get_hyperparam_value, 408
 get_hyperparams, 408
 get_image_from_stack, 408
 get_lbound, 408
 get_max_sigma, 408, 409
 get_max_sigma_ratio, 409
 get_mcmc_num_phases, 409
 get_mcmc_sigma_scale, 409
 get_min_sigma, 409
 get_num_hyperparams, 409
 get_num_params, 410
 get_num_pixels, 410
 get_param_names, 410
 get_prior, 410
 get_rng_generator, 410
 get_rng_manager, 410
 get_size, 411
 get_size_image_stack, 411
 get_stats, 411
 get_ubound, 411
 global_default_mcmc_sigma_scale, 425
 global_max_mcmc_sigma_scale, 426
 global_max_psf_sigma, 426
 global_max_size, 426
 global_min_psf_sigma, 426
 global_min_size, 426
 has_hyperparam, 411
 ImageCoordT, 402
 ImagePixelT, 403
 ImageShapeT, 403
 ImageSizeShapeT, 403
 ImageSizeVecShapeT, 403
 ImageSizeVecT, 403
 ImageSizeT, 403
 ImageStackShapeT, 403
 ImageStackT, 404
 ImageT, 404
 initial_theta_estimate, 411, 412
 lbound, 426
 make_default_prior, 412
 make_default_prior_beta_position, 412
 make_default_prior_normal_position, 412
 make_image, 413
 make_image_stack, 413
 make_internal_1Dsum_estimator, 413
 make_param, 413
 make_param_mat, 414
 make_param_mat_stack, 414
 make_param_stack, 414
 make_prior_beta_position, 414
 make_prior_component_intensity, 415
 make_prior_component_position_beta, 415
 make_prior_component_position_normal, 415
 make_prior_component_sigma, 415
 make_prior_normal_position, 416
 make_stencil, 416
 min_sigma, 427
 ModelDataStackT, 404
 ModelDataT, 404
 name, 427
 num_dim, 427
 num_hyperparams, 427
 num_params, 427
 num_phases, 427
 operator=, 416, 417
 ParamVecT, 404
 ParamT, 404
 pixel_grad, 417
 pixel_grad2, 417
 pixel_hess, 417
 pixel_hess_update, 417
 pixel_model_value, 418
 prior, 428
 prior_types, 428
 reflected_theta, 418
 reflected_theta_stack, 418

- rename_hyperparam, 418
- sample_mcmc_candidate, 418
- sample_prior, 419
- set_background_mcmc_sampling, 419
- set_bounds, 419
- set_hyperparam_names, 419
- set_hyperparam_value, 419
- set_hyperparams, 420
- set_image_in_stack, 420
- set_intensity_mcmc_sampling, 420
- set_lbound, 420
- set_max_sigma, 420
- set_max_sigma_ratio, 420
- set_mcmc_num_phases, 421
- set_mcmc_sigma_scale, 421
- set_min_sigma, 421
- set_param_names, 421
- set_prior, 421
- set_rng_seed, 422
- set_size, 422
- set_ubound, 422
- sigma_scale, 428
- size, 428
- StencilVecT, 404
- theta_in_bounds, 422
- theta_stack_in_bounds, 422
- ubound, 429
- update_internal_1Dsum_estimators, 422
- x_model, 429
- y_model, 429
- mappel::Gauss2DsModel, 430
 - bound_theta, 437
 - bounded_theta, 437
 - bounded_theta_stack, 437
 - bounds_epsilon, 454
 - check_image_shape, 438
 - check_param_shape, 438
 - check_psf_sigma, 438
 - check_size, 439
 - compute_max_sigma_ratio, 439
 - debug_internal_sum_model_x, 439
 - debug_internal_sum_model_y, 439
 - default_alpha_sigma, 454
 - default_beta_pos, 454
 - default_intensity_kappa, 454
 - default_max_l, 454
 - default_mean_l, 454
 - default_pixel_mean_bg, 455
 - default_sigma_pos, 455
 - DefaultPriorType, 455
 - DefaultSeperableInitEstimator, 455
 - eta_bg, 455
 - eta_l, 455
 - eta_sigma, 456
 - eta_x, 456
 - eta_y, 456
 - Gauss1DSumModelT, 435
 - Gauss2DsModel, 437
 - get_hyperparam_index, 439
 - get_hyperparam_names, 439
 - get_hyperparam_value, 440
 - get_hyperparams, 440
 - get_image_from_stack, 440
 - get_lbound, 440
 - get_max_sigma, 440
 - get_max_sigma_ratio, 441
 - get_mcmc_num_phases, 441
 - get_mcmc_sigma_scale, 441
 - get_min_sigma, 441
 - get_num_hyperparams, 441
 - get_num_params, 441
 - get_num_pixels, 441
 - get_param_names, 442
 - get_prior, 442
 - get_rng_generator, 442
 - get_rng_manager, 442
 - get_size, 442
 - get_size_image_stack, 443
 - get_stats, 443
 - get_ubound, 443
 - global_default_mcmc_sigma_scale, 456
 - global_max_mcmc_sigma_scale, 456
 - global_max_psf_sigma, 457
 - global_max_size, 457
 - global_min_psf_sigma, 457
 - global_min_size, 457
 - has_hyperparam, 443
 - ImageCoordT, 435
 - ImagePixelT, 435
 - ImageShapeT, 435
 - ImageSizeShapeT, 435
 - ImageSizeVecShapeT, 435
 - ImageSizeVecT, 436
 - ImageSizeT, 435
 - ImageStackShapeT, 436
 - ImageStackT, 436
 - ImageT, 436
 - initial_theta_estimate, 443, 444
 - lbound, 457
 - make_default_prior, 444
 - make_default_prior_beta_position, 444
 - make_default_prior_normal_position, 444
 - make_image, 444
 - make_image_stack, 445
 - make_internal_1Dsum_estimator, 445
 - make_param, 445
 - make_param_mat, 445
 - make_param_mat_stack, 446

- make_param_stack, 446
- make_prior_beta_position, 446
- make_prior_component_intensity, 446
- make_prior_component_position_beta, 447
- make_prior_component_position_normal, 447
- make_prior_component_sigma, 447
- make_prior_normal_position, 447
- make_stencil, 447
- min_sigma, 457
- num_dim, 458
- num_hyperparams, 458
- num_params, 458
- num_phases, 458
- operator=, 448
- ParamVecT, 436
- ParamT, 436
- pixel_grad, 448
- pixel_grad2, 448
- pixel_hess, 449
- pixel_hess_update, 449
- pixel_model_value, 449
- prior, 458
- prior_types, 459
- reflected_theta, 449
- reflected_theta_stack, 449
- rename_hyperparam, 450
- sample_mcmc_candidate, 450
- sample_prior, 450
- set_background_mcmc_sampling, 450
- set_bounds, 450
- set_hyperparam_names, 451
- set_hyperparam_value, 451
- set_hyperparams, 451
- set_image_in_stack, 451
- set_intensity_mcmc_sampling, 451
- set_lbound, 451
- set_max_sigma, 452
- set_max_sigma_ratio, 452
- set_mcmc_num_phases, 452
- set_mcmc_sigma_scale, 452
- set_min_sigma, 452
- set_param_names, 452
- set_prior, 453
- set_rng_seed, 453
- set_size, 453
- set_ubound, 453
- sigma_scale, 459
- size, 459
- StencilVecT, 436
- theta_in_bounds, 453
- theta_stack_in_bounds, 453
- ubound, 460
- update_internal_1Dsum_estimators, 454
- x_model, 460
- y_model, 460
- mappel::Gauss2DsModel::Stencil, 758
 - bg, 760
 - compute_derivatives, 760
 - DXS2, 762
 - DXSX, 762
 - DXS, 762
 - DYS2, 763
 - DYSY, 763
 - DYS, 762
 - derivatives_computed, 761
 - DX, 762
 - dx, 761
 - DY, 762
 - dy, 762
 - Gx, 763
 - Gy, 763
 - I, 760
 - model, 763
 - operator<<, 761
 - ParamT, 759
 - sigma_ratio, 760
 - sigmaX, 760
 - sigmaY, 761
 - Stencil, 759
 - theta, 763
 - X, 763
 - x, 761
 - Y, 764
 - y, 761
- mappel::Gauss2DsxyMAP, 461
 - bound_theta, 467
 - bounded_theta, 467
 - bounded_theta_stack, 468
 - bounds_epsilon, 482
 - check_image_shape, 468
 - check_param_shape, 468
 - check_psf_sigma, 468, 469
 - check_size, 469
 - default_alpha_sigma, 482
 - default_beta_pos, 482
 - default_intensity_kappa, 482
 - default_max_I, 482
 - default_mean_I, 483
 - default_pixel_mean_bg, 483
 - default_sigma_pos, 483
 - DefaultPriorType, 483
 - DefaultSeperableInitEstimator, 483
 - estimator_names, 483
 - eta_bg, 484
 - eta_I, 484
 - eta_x, 484
 - Gauss2DsxyMAP, 467
 - get_hyperparam_index, 469

get_hyperparam_names, 469
get_hyperparam_value, 469
get_hyperparams, 469
get_image_from_stack, 470
get_lbound, 470
get_mcmc_num_phases, 470
get_mcmc_sigma_scale, 470
get_num_hyperparams, 470
get_num_params, 470
get_num_pixels, 470
get_param_names, 471
get_prior, 471
get_psf_sigma, 471
get_rng_generator, 471
get_rng_manager, 471
get_size, 472
get_size_image_stack, 472
get_stats, 472
get_ubound, 472
global_default_mcmc_sigma_scale, 484
global_max_mcmc_sigma_scale, 484
global_max_psf_sigma, 484
global_max_size, 485
global_min_psf_sigma, 485
global_min_size, 485
has_hyperparam, 472
ImageCoordT, 465
ImagePixelT, 465
ImageShapeT, 465
ImageSizeShapeT, 465
ImageSizeVecShapeT, 466
ImageSizeVecT, 466
ImageSizeT, 466
ImageStackShapeT, 466
ImageStackT, 466
ImageT, 466
initial_theta_estimate, 472, 473
lbound, 485
make_default_prior, 473
make_default_prior_beta_position, 473
make_default_prior_normal_position, 473
make_image, 473
make_image_stack, 473
make_param, 474
make_param_mat, 474
make_param_mat_stack, 474
make_param_stack, 474, 475
make_prior_beta_position, 475
make_prior_component_intensity, 475
make_prior_component_position_beta, 475
make_prior_component_position_normal, 475
make_prior_component_sigma, 476
make_prior_normal_position, 476
make_stencil, 476
ModelDataStackT, 466
ModelDataT, 467
name, 485
num_dim, 485
num_hyperparams, 486
num_params, 486
num_phases, 486
ParamVecT, 467
ParamT, 467
pixel_grad, 477
pixel_grad2, 477
pixel_hess, 477
pixel_hess_update, 477
pixel_model_value, 477
prior, 486
prior_types, 486
psf_sigma, 487
reflected_theta, 477
reflected_theta_stack, 478
rename_hyperparam, 478
sample_mcmc_candidate, 478
sample_prior, 478
set_background_mcmc_sampling, 478
set_bounds, 479
set_hyperparam_names, 479
set_hyperparam_value, 479
set_hyperparams, 479
set_image_in_stack, 479
set_intensity_mcmc_sampling, 479
set_lbound, 480
set_mcmc_num_phases, 480
set_mcmc_sigma_scale, 480
set_param_names, 480
set_prior, 480
set_psf_sigma, 481
set_rng_seed, 481
set_size, 481
set_ubound, 481
sigma_scale, 487
size, 487
StencilVecT, 467
theta_in_bounds, 481
theta_stack_in_bounds, 482
ubound, 487
mappel::Gauss2DsxyModel, 488
bound_theta, 494
bounded_theta, 494
bounded_theta_stack, 494
bounds_epsilon, 506
check_image_shape, 494
check_param_shape, 494, 495
check_psf_sigma, 495
check_size, 495
compute_max_sigma_ratio, 495

default_alpha_sigma, 506
default_beta_pos, 506
default_intensity_kappa, 507
default_max_l, 507
default_mean_l, 507
default_pixel_mean_bg, 507
default_sigma_pos, 507
DefaultSeperableInitEstimator, 507
Gauss2DsxyModel, 494
get_hyperparam_index, 496
get_hyperparam_names, 496
get_hyperparam_value, 496
get_hyperparams, 496
get_image_from_stack, 496
get_lbound, 496
get_max_sigma, 496, 497
get_max_sigma_ratio, 497
get_min_sigma, 497
get_num_hyperparams, 497
get_num_params, 497
get_num_pixels, 497
get_param_names, 497
get_prior, 498
get_rng_generator, 498
get_rng_manager, 498
get_size, 498
get_size_image_stack, 498
get_stats, 499
get_ubound, 499
global_max_psf_sigma, 508
global_max_size, 508
global_min_psf_sigma, 508
global_min_size, 508
has_hyperparam, 499
ImageCoordT, 492
ImagePixelT, 492
ImageShapeT, 492
ImageSizeShapeT, 492
ImageSizeVecShapeT, 492
ImageSizeVecT, 493
ImageSizeT, 492
ImageStackShapeT, 493
ImageStackT, 493
ImageT, 493
initial_theta_estimate, 499
lbound, 508
make_default_prior, 499
make_image, 499
make_image_stack, 499
make_param, 500
make_param_mat, 500
make_param_mat_stack, 500
make_param_stack, 500, 501
make_prior_beta_position, 501
make_prior_component_intensity, 501
make_prior_component_position_beta, 501
make_prior_component_position_normal, 501
make_prior_component_sigma, 502
make_prior_normal_position, 502
make_stencil, 502
mcmc_candidate_eta_sigma, 509
mcmc_candidate_eta_y, 509
min_sigma, 509
num_dim, 509
num_hyperparams, 509
num_params, 509
ParamVecT, 493
ParamT, 493
pixel_grad, 503
pixel_grad2, 503
pixel_hess, 503
pixel_hess_update, 503
pixel_model_value, 503
prior, 510
reflected_theta, 503
reflected_theta_stack, 503
rename_hyperparam, 504
sample_mcmc_candidate, 504
sample_prior, 504
set_bounds, 504
set_hyperparam_names, 504
set_hyperparam_value, 504
set_hyperparams, 505
set_image_in_stack, 505
set_lbound, 505
set_max_sigma, 505
set_max_sigma_ratio, 505
set_min_sigma, 505
set_param_names, 505
set_prior, 505
set_rng_seed, 505
set_size, 505
set_ubound, 506
size, 510
StencilVecT, 493
theta_in_bounds, 506
theta_stack_in_bounds, 506
ubound, 511
update_internal_1D_estimators, 506
x_model, 511
y_model, 511
mappel::Gauss2DsxyModel::Stencil, 753
bg, 755
compute_derivatives, 755
DXS2, 756
DXSX, 756
DXS, 756
DYS2, 757

DYSX, [757](#)
 DYSY, [757](#)
 DYS, [757](#)
 derivatives_computed, [756](#)
 DX, [756](#)
 dx, [756](#)
 DY, [757](#)
 dy, [757](#)
 Gx, [757](#)
 Gy, [757](#)
 I, [755](#)
 model, [757](#)
 operator<<, [756](#)
 ParamT, [754](#)
 sigmaX, [755](#)
 sigmaY, [755](#)
 Stencil, [755](#)
 theta, [758](#)
 X, [758](#)
 x, [755](#)
 Y, [758](#)
 y, [756](#)
 mappel::HeuristicEstimator
 clear_stats, [514](#)
 compute_estimate, [514](#)
 compute_estimate_debug, [514](#)
 compute_profile_estimate, [514](#)
 Error, [513](#)
 estimate_max, [514](#), [515](#)
 estimate_max_debug, [515](#)
 estimate_max_stack, [516](#)
 estimate_profile_stack, [516](#)
 exit_counts, [518](#)
 ExitCode, [513](#)
 FunctionChange, [513](#)
 get_debug_stats, [516](#)
 get_model, [516](#)
 get_stats, [517](#)
 GradRatio, [513](#)
 HeuristicEstimator, [513](#)
 max_threads, [518](#)
 MaxBacktracks, [513](#)
 MaxIter, [513](#)
 model, [518](#)
 mtx, [518](#)
 name, [517](#)
 num_estimations, [518](#)
 num_threads, [519](#)
 NumExitCodes, [519](#)
 record_exit_code, [517](#)
 record_walltime, [517](#)
 set_model, [517](#)
 StepSize, [513](#)
 Success, [513](#)
 total_walltime, [519](#)
 TrustRegionRadius, [513](#)
 Unassigned, [513](#)
 mappel::HeuristicEstimator< Model >, [511](#)
 mappel::ImageFormat1DBase, [519](#)
 check_image_shape, [523](#)
 check_size, [523](#)
 get_image_from_stack, [523](#)
 get_num_pixels, [523](#)
 get_size, [523](#)
 get_size_image_stack, [524](#)
 get_stats, [524](#)
 global_max_size, [525](#)
 global_min_size, [525](#)
 ImageCoordT, [521](#)
 ImageFormat1DBase, [522](#)
 ImagePixelT, [521](#)
 ImageShapeT, [521](#)
 ImageSizeShapeT, [521](#)
 ImageSizeVecShapeT, [522](#)
 ImageSizeVecT, [522](#)
 ImageSizeT, [521](#)
 ImageStackShapeT, [522](#)
 ImageStackT, [522](#)
 ImageT, [522](#)
 make_image, [524](#)
 make_image_stack, [524](#)
 num_dim, [525](#)
 set_image_in_stack, [524](#)
 set_size, [524](#)
 size, [525](#)
 mappel::ImageFormat2DBase, [526](#)
 check_image_shape, [529](#)
 check_size, [529](#)
 get_image_from_stack, [529](#)
 get_num_pixels, [530](#)
 get_size, [530](#)
 get_size_image_stack, [530](#)
 get_stats, [530](#)
 global_max_size, [531](#)
 global_min_size, [531](#)
 ImageCoordT, [527](#)
 ImageFormat2DBase, [529](#)
 ImagePixelT, [527](#)
 ImageShapeT, [527](#)
 ImageSizeShapeT, [528](#)
 ImageSizeVecShapeT, [528](#)
 ImageSizeVecT, [528](#)
 ImageSizeT, [528](#)
 ImageStackShapeT, [528](#)
 ImageStackT, [528](#)
 ImageT, [528](#)
 make_image, [530](#)
 make_image_stack, [530](#)

- num_dim, [531](#)
- operator=, [530](#), [531](#)
- set_image_in_stack, [531](#)
- set_size, [531](#)
- size, [532](#)
- mappel::IterativeMaximizer
 - alpha, [541](#)
 - backtrack, [535](#)
 - clear_stats, [535](#)
 - compute_estimate, [536](#)
 - compute_estimate_debug, [536](#)
 - compute_profile_estimate, [536](#)
 - convergence_test, [537](#)
 - delta, [541](#)
 - epsilon, [541](#)
 - Error, [535](#)
 - estimate_max, [537](#), [538](#)
 - estimate_max_debug, [538](#)
 - estimate_max_stack, [538](#)
 - estimate_profile_stack, [539](#)
 - exit_counts, [542](#)
 - ExitCode, [535](#)
 - FunctionChange, [535](#)
 - get_debug_stats, [539](#)
 - get_model, [539](#)
 - get_stats, [539](#)
 - GradRatio, [535](#)
 - IterativeMaximizer, [535](#)
 - lambda_min, [542](#)
 - last_backtrack_idx, [542](#)
 - local_maximize, [539](#)
 - max_backtracks, [542](#)
 - max_iterations, [542](#)
 - max_threads, [542](#)
 - MaxBacktracks, [535](#)
 - MaxIter, [535](#)
 - maximize, [540](#)
 - mean_backtracks, [540](#)
 - mean_der_evals, [540](#)
 - mean_fun_evals, [540](#)
 - mean_iterations, [540](#)
 - model, [542](#)
 - mtx, [543](#)
 - name, [540](#)
 - num_estimations, [543](#)
 - num_threads, [543](#)
 - NumExitCodes, [543](#)
 - record_exit_code, [540](#)
 - record_run_statistics, [540](#)
 - record_walltime, [541](#)
 - set_model, [541](#)
 - StepSize, [535](#)
 - Success, [535](#)
 - total_backtracks, [543](#)
 - total_der_evals, [544](#)
 - total_fun_evals, [544](#)
 - total_iterations, [544](#)
 - total_walltime, [544](#)
 - TrustRegionRadius, [535](#)
 - Unassigned, [535](#)
- mappel::IterativeMaximizer< Model >, [532](#)
- mappel::IterativeMaximizer< Model >::MaximizerData, [560](#)
- mappel::IterativeMaximizer::MaximizerData
 - backtrack_idx, [565](#)
 - current_stencil, [565](#)
 - fixed_idx, [565](#)
 - free_parameters, [565](#)
 - get_backtrack_idx, [562](#)
 - get_theta_sequence, [562](#)
 - get_theta_sequence_rllh, [562](#)
 - getIteration, [562](#)
 - grad, [565](#)
 - im, [565](#)
 - lbound, [565](#)
 - max_seq_len, [565](#)
 - MaximizerData, [561](#)
 - nBacktracks, [566](#)
 - nIterations, [566](#)
 - num_fixed_parameters, [566](#)
 - record_backtrack, [562](#)
 - record_iteration, [562](#), [563](#)
 - restore_stencil, [563](#)
 - rllh, [566](#)
 - s0, [566](#)
 - s1, [566](#)
 - save_seq, [566](#)
 - save_stencil, [563](#)
 - saved_stencil, [563](#)
 - saved_theta, [563](#)
 - seq_len, [567](#)
 - seq_rllh, [567](#)
 - set_fixed_parameters, [564](#)
 - set_stencil, [564](#)
 - stencil, [564](#)
 - step, [567](#)
 - theta, [564](#)
 - theta_seq, [567](#)
 - ubound, [567](#)
- mappel::LogicalError, [544](#)
- LogicalError, [545](#)
- mappel::MAPEstimator, [545](#)
- bound_theta, [548](#)
- bounded_theta, [548](#)
- bounded_theta_stack, [549](#)
- bounds_epsilon, [557](#)
- check_param_shape, [549](#)
- check_psf_sigma, [549](#)

default_alpha_sigma, 557
default_beta_pos, 557
default_intensity_kappa, 557
default_max_l, 557
default_mean_l, 558
default_pixel_mean_bg, 558
default_sigma_pos, 558
DefaultSeperableInitEstimator, 558
get_hyperparam_index, 549
get_hyperparam_names, 550
get_hyperparam_value, 550
get_hyperparams, 550
get_lbound, 550
get_num_hyperparams, 550
get_num_params, 550
get_param_names, 550
get_prior, 551
get_rng_generator, 551
get_rng_manager, 551
get_stats, 551
get_ubound, 551
global_max_psf_sigma, 558
global_min_psf_sigma, 558
has_hyperparam, 551
lbound, 559
MAPEstimator, 548
make_param, 552
make_param_mat, 552
make_param_mat_stack, 552
make_param_stack, 552, 553
make_prior_component_intensity, 553
make_prior_component_position_beta, 553
make_prior_component_position_normal, 553
make_prior_component_sigma, 553
num_hyperparams, 559
num_params, 559
operator=, 554
ParamVecT, 548
ParamT, 548
prior, 559
reflected_theta, 554
reflected_theta_stack, 554
rename_hyperparam, 554
sample_prior, 554, 555
set_bounds, 555
set_hyperparam_names, 555
set_hyperparam_value, 555
set_hyperparams, 555
set_lbound, 555
set_param_names, 556
set_prior, 556
set_rng_seed, 556
set_ubound, 556
theta_in_bounds, 556
theta_stack_in_bounds, 557
ubound, 559
mappel::MCMCAdaptor1Ds, 586
bound_theta, 589
bounded_theta, 589
bounded_theta_stack, 590
bounds_epsilon, 599
check_param_shape, 590
check_psf_sigma, 590
default_alpha_sigma, 599
default_beta_pos, 599
default_intensity_kappa, 599
default_max_l, 600
default_mean_l, 600
default_pixel_mean_bg, 600
default_sigma_pos, 600
DefaultSeperableInitEstimator, 600
eta_bg, 600
eta_l, 601
eta_sigma, 601
eta_x, 601
get_hyperparam_index, 590
get_hyperparam_names, 591
get_hyperparam_value, 591
get_hyperparams, 591
get_lbound, 591
get_mcmc_num_phases, 591
get_mcmc_sigma_scale, 591
get_num_hyperparams, 591
get_num_params, 592
get_param_names, 592
get_prior, 592
get_rng_generator, 592
get_rng_manager, 592
get_stats, 592
get_ubound, 592
global_default_mcmc_sigma_scale, 601
global_max_mcmc_sigma_scale, 601
global_max_psf_sigma, 601
global_min_psf_sigma, 602
has_hyperparam, 593
lbound, 602
MCMCAdaptor1Ds, 589
make_param, 593
make_param_mat, 593
make_param_mat_stack, 593, 594
make_param_stack, 594
make_prior_component_intensity, 594
make_prior_component_position_beta, 594
make_prior_component_position_normal, 595
make_prior_component_sigma, 595
num_hyperparams, 602
num_params, 602
num_phases, 602

- operator=, 595
- ParamVecT, 589
- ParamT, 589
- prior, 603
- reflected_theta, 595
- reflected_theta_stack, 596
- rename_hyperparam, 596
- sample_mcmc_candidate, 596
- sample_prior, 596
- set_background_mcmc_sampling, 596
- set_bounds, 597
- set_hyperparam_names, 597
- set_hyperparam_value, 597
- set_hyperparams, 597
- set_intensity_mcmc_sampling, 597
- set_lbound, 597
- set_mcmc_num_phases, 598
- set_mcmc_sigma_scale, 598
- set_param_names, 598
- set_prior, 598
- set_rng_seed, 598
- set_ubound, 598
- sigma_scale, 603
- theta_in_bounds, 599
- theta_stack_in_bounds, 599
- ubound, 603
- mappel::MCMCAdaptor1D, 568
 - bound_theta, 571
 - bounded_theta, 571
 - bounded_theta_stack, 571
 - bounds_epsilon, 581
 - check_param_shape, 572
 - check_psf_sigma, 572
 - default_alpha_sigma, 581
 - default_beta_pos, 581
 - default_intensity_kappa, 582
 - default_max_l, 582
 - default_mean_l, 582
 - default_pixel_mean_bg, 582
 - default_sigma_pos, 582
 - DefaultSeperableInitEstimator, 582
 - eta_bg, 583
 - eta_l, 583
 - eta_x, 583
 - get_hyperparam_index, 572
 - get_hyperparam_names, 572
 - get_hyperparam_value, 573
 - get_hyperparams, 573
 - get_lbound, 573
 - get_mcmc_num_phases, 573
 - get_mcmc_sigma_scale, 573
 - get_num_hyperparams, 573
 - get_num_params, 573
 - get_param_names, 574
 - get_prior, 574
 - get_rng_generator, 574
 - get_rng_manager, 574
 - get_stats, 574
 - get_ubound, 574
 - global_default_mcmc_sigma_scale, 583
 - global_max_mcmc_sigma_scale, 583
 - global_max_psf_sigma, 583
 - global_min_psf_sigma, 584
 - has_hyperparam, 575
 - lbound, 584
 - MCMCAdaptor1D, 571
 - make_param, 575
 - make_param_mat, 575
 - make_param_mat_stack, 575, 576
 - make_param_stack, 576
 - make_prior_component_intensity, 576
 - make_prior_component_position_beta, 576
 - make_prior_component_position_normal, 577
 - make_prior_component_sigma, 577
 - num_hyperparams, 584
 - num_params, 584
 - num_phases, 584
 - operator=, 577
 - ParamVecT, 570
 - ParamT, 570
 - prior, 585
 - reflected_theta, 577
 - reflected_theta_stack, 578
 - rename_hyperparam, 578
 - sample_mcmc_candidate, 578
 - sample_prior, 578
 - set_background_mcmc_sampling, 578
 - set_bounds, 579
 - set_hyperparam_names, 579
 - set_hyperparam_value, 579
 - set_hyperparams, 579
 - set_intensity_mcmc_sampling, 579
 - set_lbound, 579
 - set_mcmc_num_phases, 580
 - set_mcmc_sigma_scale, 580
 - set_param_names, 580
 - set_prior, 580
 - set_rng_seed, 580
 - set_ubound, 581
 - sigma_scale, 585
 - theta_in_bounds, 581
 - theta_stack_in_bounds, 581
 - ubound, 585
- mappel::MCMCAdaptor2Ds, 622
 - bound_theta, 626
 - bounded_theta, 626
 - bounded_theta_stack, 626
 - bounds_epsilon, 636

[check_param_shape](#), 626
[check_psf_sigma](#), 627
[default_alpha_sigma](#), 636
[default_beta_pos](#), 636
[default_intensity_kappa](#), 636
[default_max_l](#), 636
[default_mean_l](#), 636
[default_pixel_mean_bg](#), 636
[default_sigma_pos](#), 637
[DefaultSeperableInitEstimator](#), 637
[eta_bg](#), 637
[eta_l](#), 637
[eta_sigma](#), 637
[eta_x](#), 638
[eta_y](#), 638
[get_hyperparam_index](#), 627
[get_hyperparam_names](#), 627
[get_hyperparam_value](#), 627
[get_hyperparams](#), 627
[get_lbound](#), 628
[get_mcmc_num_phases](#), 628
[get_mcmc_sigma_scale](#), 628
[get_num_hyperparams](#), 628
[get_num_params](#), 628
[get_param_names](#), 628
[get_prior](#), 628, 629
[get_rng_generator](#), 629
[get_rng_manager](#), 629
[get_stats](#), 629
[get_ubound](#), 629
[global_default_mcmc_sigma_scale](#), 638
[global_max_mcmc_sigma_scale](#), 638
[global_max_psf_sigma](#), 638
[global_min_psf_sigma](#), 638
[has_hyperparam](#), 629
[lbound](#), 639
[MCMCAdaptor2Ds](#), 625, 626
[make_param](#), 629, 630
[make_param_mat](#), 630
[make_param_mat_stack](#), 630
[make_param_stack](#), 630
[make_prior_component_intensity](#), 631
[make_prior_component_position_beta](#), 631
[make_prior_component_position_normal](#), 631
[make_prior_component_sigma](#), 631
[num_hyperparams](#), 639
[num_params](#), 639
[num_phases](#), 639
[operator=](#), 631, 632
[ParamVecT](#), 625
[ParamT](#), 625
[prior](#), 639
[reflected_theta](#), 632
[reflected_theta_stack](#), 632
[rename_hyperparam](#), 632
[sample_mcmc_candidate](#), 632
[sample_prior](#), 632, 633
[set_background_mcmc_sampling](#), 633
[set_bounds](#), 633
[set_hyperparam_names](#), 633
[set_hyperparam_value](#), 633
[set_hyperparams](#), 633
[set_intensity_mcmc_sampling](#), 634
[set_lbound](#), 634
[set_mcmc_num_phases](#), 634
[set_mcmc_sigma_scale](#), 634
[set_param_names](#), 634
[set_prior](#), 634, 635
[set_rng_seed](#), 635
[set_ubound](#), 635
[sigma_scale](#), 640
[theta_in_bounds](#), 635
[theta_stack_in_bounds](#), 635
[ubound](#), 640
[mappel::MCMCAdaptor2D](#), 604
[bound_theta](#), 607
[bounded_theta](#), 607
[bounded_theta_stack](#), 608
[bounds_epsilon](#), 617
[check_param_shape](#), 608
[check_psf_sigma](#), 608
[default_alpha_sigma](#), 617
[default_beta_pos](#), 617
[default_intensity_kappa](#), 618
[default_max_l](#), 618
[default_mean_l](#), 618
[default_pixel_mean_bg](#), 618
[default_sigma_pos](#), 618
[DefaultSeperableInitEstimator](#), 618
[eta_bg](#), 619
[eta_l](#), 619
[eta_x](#), 619
[eta_y](#), 619
[get_hyperparam_index](#), 608
[get_hyperparam_names](#), 609
[get_hyperparam_value](#), 609
[get_hyperparams](#), 609
[get_lbound](#), 609
[get_mcmc_num_phases](#), 609
[get_mcmc_sigma_scale](#), 609
[get_num_hyperparams](#), 609
[get_num_params](#), 610
[get_param_names](#), 610
[get_prior](#), 610
[get_rng_generator](#), 610
[get_rng_manager](#), 610
[get_stats](#), 610
[get_ubound](#), 610

- global_default_mcmc_sigma_scale, 619
- global_max_mcmc_sigma_scale, 620
- global_max_psf_sigma, 620
- global_min_psf_sigma, 620
- has_hyperparam, 611
- lbound, 620
- MCMCAdaptor2D, 607
- make_param, 611
- make_param_mat, 611
- make_param_mat_stack, 611, 612
- make_param_stack, 612
- make_prior_component_intensity, 612
- make_prior_component_position_beta, 612
- make_prior_component_position_normal, 613
- make_prior_component_sigma, 613
- num_hyperparams, 620
- num_params, 620
- num_phases, 621
- operator=, 613
- ParamVecT, 607
- ParamT, 607
- prior, 621
- reflected_theta, 613
- reflected_theta_stack, 614
- rename_hyperparam, 614
- sample_mcmc_candidate, 614
- sample_prior, 614
- set_background_mcmc_sampling, 614
- set_bounds, 615
- set_hyperparam_names, 615
- set_hyperparam_value, 615
- set_hyperparams, 615
- set_intensity_mcmc_sampling, 615
- set_lbound, 615
- set_mcmc_num_phases, 616
- set_mcmc_sigma_scale, 616
- set_param_names, 616
- set_prior, 616
- set_rng_seed, 616
- set_ubound, 617
- sigma_scale, 621
- theta_in_bounds, 617
- theta_stack_in_bounds, 617
- ubound, 621
- mappel::MCMCAdaptorBase, 641
 - get_mcmc_num_phases, 642
 - get_mcmc_sigma_scale, 642
 - get_stats, 642
 - global_default_mcmc_sigma_scale, 643
 - global_max_mcmc_sigma_scale, 643
 - MCMCAdaptorBase, 641
 - num_phases, 643
 - set_mcmc_num_phases, 642
 - set_mcmc_sigma_scale, 642
 - sigma_scale, 643
- mappel::MLEstimator, 643
 - bound_theta, 646
 - bounded_theta, 646
 - bounded_theta_stack, 647
 - bounds_epsilon, 655
 - check_param_shape, 647
 - check_psf_sigma, 647
 - default_alpha_sigma, 655
 - default_beta_pos, 655
 - default_intensity_kappa, 655
 - default_max_l, 655
 - default_mean_l, 656
 - default_pixel_mean_bg, 656
 - default_sigma_pos, 656
 - DefaultSeperableInitEstimator, 656
 - get_hyperparam_index, 647
 - get_hyperparam_names, 648
 - get_hyperparam_value, 648
 - get_hyperparams, 648
 - get_lbound, 648
 - get_num_hyperparams, 648
 - get_num_params, 648
 - get_param_names, 648
 - get_prior, 649
 - get_rng_generator, 649
 - get_rng_manager, 649
 - get_stats, 649
 - get_ubound, 649
 - global_max_psf_sigma, 656
 - global_min_psf_sigma, 656
 - has_hyperparam, 649
 - lbound, 657
 - MLEstimator, 646
 - make_param, 650
 - make_param_mat, 650
 - make_param_mat_stack, 650
 - make_param_stack, 650, 651
 - make_prior_component_intensity, 651
 - make_prior_component_position_beta, 651
 - make_prior_component_position_normal, 651
 - make_prior_component_sigma, 651
 - num_hyperparams, 657
 - num_params, 657
 - operator=, 652
 - ParamVecT, 646
 - ParamT, 646
 - prior, 657
 - reflected_theta, 652
 - reflected_theta_stack, 652
 - rename_hyperparam, 652
 - sample_prior, 652, 653
 - set_bounds, 653
 - set_hyperparam_names, 653

- set_hyperparam_value, 653
- set_hyperparams, 653
- set_lbound, 653
- set_param_names, 654
- set_prior, 654
- set_rng_seed, 654
- set_ubound, 654
- theta_in_bounds, 654
- theta_stack_in_bounds, 655
- ubound, 657
- mappel::ModelBoundsError, 658
 - ModelBoundsError, 658
- mappel::NewtonDiagonalMaximizer
 - alpha, 668
 - backtrack, 662
 - clear_stats, 662
 - compute_estimate, 662
 - compute_estimate_debug, 662
 - compute_profile_estimate, 663
 - convergence_test, 663
 - delta, 668
 - epsilon, 668
 - Error, 661
 - estimate_max, 663, 664
 - estimate_max_debug, 664
 - estimate_max_stack, 664, 665
 - estimate_profile_stack, 665
 - exit_counts, 668
 - ExitCode, 661
 - FunctionChange, 661
 - get_debug_stats, 665
 - get_model, 665
 - get_stats, 665
 - GradRatio, 661
 - lambda_min, 669
 - last_backtrack_idx, 669
 - local_maximize, 666
 - max_backtracks, 669
 - max_iterations, 669
 - max_threads, 669
 - MaxBacktracks, 661
 - MaxIter, 661
 - maximize, 666
 - MaximizerData, 661
 - mean_backtracks, 666
 - mean_der_evals, 667
 - mean_fun_evals, 667
 - mean_iterations, 667
 - model, 669
 - mtx, 670
 - name, 667
 - NewtonDiagonalMaximizer, 661
 - num_estimations, 670
 - num_threads, 670
 - NumExitCodes, 670
 - record_exit_code, 667
 - record_run_statistics, 667
 - record_walltime, 667
 - set_model, 668
 - StepSize, 661
 - Success, 661
 - total_backtracks, 670
 - total_der_evals, 671
 - total_fun_evals, 671
 - total_iterations, 671
 - total_walltime, 671
 - TrustRegionRadius, 661
 - Unassigned, 661
- mappel::NewtonDiagonalMaximizer< Model >, 659
- mappel::NewtonMaximizer
 - alpha, 681
 - backtrack, 675
 - clear_stats, 675
 - compute_estimate, 675
 - compute_estimate_debug, 675
 - compute_profile_estimate, 676
 - convergence_test, 676
 - delta, 681
 - epsilon, 681
 - Error, 674
 - estimate_max, 676, 677
 - estimate_max_debug, 677
 - estimate_max_stack, 677, 678
 - estimate_profile_stack, 678
 - exit_counts, 681
 - ExitCode, 674
 - FunctionChange, 674
 - get_debug_stats, 678
 - get_model, 678
 - get_stats, 678
 - GradRatio, 674
 - lambda_min, 682
 - last_backtrack_idx, 682
 - local_maximize, 679
 - max_backtracks, 682
 - max_iterations, 682
 - max_threads, 682
 - MaxBacktracks, 674
 - MaxIter, 674
 - maximize, 679
 - MaximizerData, 674
 - mean_backtracks, 679
 - mean_der_evals, 680
 - mean_fun_evals, 680
 - mean_iterations, 680
 - model, 682
 - mtx, 683
 - name, 680

- NewtonMaximizer, 674
- num_estimations, 683
- num_threads, 683
- NumExitCodes, 683
- record_exit_code, 680
- record_run_statistics, 680
- record_walltime, 680
- set_model, 681
- StepSize, 674
- Success, 674
- total_backtracks, 683
- total_der_evals, 684
- total_fun_evals, 684
- total_iterations, 684
- total_walltime, 684
- TrustRegionRadius, 674
- Unassigned, 674
- mappel::NewtonMaximizer< Model >, 672
- mappel::NotImplementedError, 685
 - NotImplementedError, 685
- mappel::NumericalError, 685
 - NumericalError, 686
- mappel::ParameterValueError, 688
 - ParameterValueError, 689
- mappel::PointEmitterModel, 689
 - bound_theta, 693
 - bounded_theta, 693
 - bounded_theta_stack, 693
 - bounds_epsilon, 701
 - check_param_shape, 693
 - check_psf_sigma, 694
 - default_alpha_sigma, 701
 - default_beta_pos, 701
 - default_intensity_kappa, 701
 - default_max_l, 702
 - default_mean_l, 702
 - default_pixel_mean_bg, 702
 - default_sigma_pos, 702
 - DefaultSeperableInitEstimator, 702
 - get_hyperparam_index, 694
 - get_hyperparam_names, 694
 - get_hyperparam_value, 694
 - get_hyperparams, 694
 - get_lbound, 694
 - get_num_hyperparams, 695
 - get_num_params, 695
 - get_param_names, 695
 - get_prior, 695
 - get_rng_generator, 695
 - get_rng_manager, 695
 - get_stats, 696
 - get_ubound, 696
 - global_max_psf_sigma, 702
 - global_min_psf_sigma, 703
 - has_hyperparam, 696
 - lbound, 703
 - make_param, 696
 - make_param_mat, 696, 697
 - make_param_mat_stack, 697
 - make_param_stack, 697
 - make_prior_component_intensity, 697
 - make_prior_component_position_beta, 697
 - make_prior_component_position_normal, 698
 - make_prior_component_sigma, 698
 - num_hyperparams, 703
 - num_params, 703
 - operator=, 698
 - ParamVecT, 692
 - ParamT, 692
 - PointEmitterModel, 692, 693
 - prior, 703
 - reflected_theta, 698
 - reflected_theta_stack, 699
 - rename_hyperparam, 699
 - sample_prior, 699
 - set_bounds, 699
 - set_hyperparam_names, 699
 - set_hyperparam_value, 699
 - set_hyperparams, 700
 - set_lbound, 700
 - set_param_names, 700
 - set_prior, 700
 - set_rng_seed, 700
 - set_ubound, 700
 - theta_in_bounds, 701
 - theta_stack_in_bounds, 701
 - ubound, 703
- mappel::PoissonNoise1DObjective, 706
 - check_image_shape, 710
 - check_size, 710
 - estimator_names, 712
 - get_image_from_stack, 710
 - get_num_pixels, 711
 - get_size, 711
 - get_size_image_stack, 711
 - get_stats, 711
 - global_max_size, 712
 - global_min_size, 713
 - ImageCoordT, 708
 - ImagePixelT, 708
 - ImageShapeT, 708
 - ImageSizeShapeT, 708
 - ImageSizeVecShapeT, 709
 - ImageSizeVecT, 709
 - ImageSizeT, 708
 - ImageStackShapeT, 709
 - ImageStackT, 709
 - ImageT, 709

- make_image, 711
- make_image_stack, 711
- ModelDataStackT, 709
- ModelDataT, 709
- num_dim, 713
- operator=, 712
- PoissonNoise1DObjective, 710
- set_image_in_stack, 712
- set_size, 712
- size, 713
- mappel::PoissonNoise2DObjective, 714
 - check_image_shape, 717
 - check_size, 718
 - estimator_names, 720
 - get_image_from_stack, 718
 - get_num_pixels, 718
 - get_size, 718
 - get_size_image_stack, 718
 - get_stats, 718
 - global_max_size, 720
 - global_min_size, 720
 - ImageCoordT, 715
 - ImagePixelT, 715
 - ImageShapeT, 715
 - ImageSizeShapeT, 716
 - ImageSizeVecShapeT, 716
 - ImageSizeVecT, 716
 - ImageSizeT, 716
 - ImageStackShapeT, 716
 - ImageStackT, 716
 - ImageT, 716
 - make_image, 719
 - make_image_stack, 719
 - ModelDataStackT, 717
 - ModelDataT, 717
 - num_dim, 720
 - operator=, 719
 - PoissonNoise2DObjective, 717
 - set_image_in_stack, 719
 - set_size, 719
 - size, 720
- mappel::PriorMAP1DObjective, 721
 - log_prior_l_const, 723
 - log_prior_bg_const, 723
 - log_prior_pos_const, 723
 - ParamMatT, 722
 - ParamT, 722
 - prior_grad2_update, 722
 - prior_grad_update, 722
 - prior_hess_update, 722
 - prior_log_likelihood, 722
 - prior_relative_log_likelihood, 723
 - set_hyperparameters, 723
- mappel::QuasiNewtonMaximizer
 - alpha, 733
 - backtrack, 727
 - clear_stats, 727
 - compute_estimate, 727
 - compute_estimate_debug, 728
 - compute_profile_estimate, 728
 - convergence_test, 728
 - delta, 733
 - epsilon, 733
 - Error, 726
 - estimate_max, 728, 729
 - estimate_max_debug, 729
 - estimate_max_stack, 730
 - estimate_profile_stack, 730
 - exit_counts, 733
 - ExitCode, 726
 - FunctionChange, 726
 - get_debug_stats, 730
 - get_model, 730
 - get_stats, 731
 - GradRatio, 726
 - lambda_min, 734
 - last_backtrack_idx, 734
 - local_maximize, 731
 - max_backtracks, 734
 - max_iterations, 734
 - max_threads, 734
 - MaxBacktracks, 726
 - MaxIter, 726
 - maximize, 731
 - MaximizerData, 726
 - mean_backtracks, 731
 - mean_der_evals, 732
 - mean_fun_evals, 732
 - mean_iterations, 732
 - model, 734
 - mtx, 735
 - name, 732
 - num_estimations, 735
 - num_threads, 735
 - NumExitCodes, 735
 - QuasiNewtonMaximizer, 727
 - record_exit_code, 732
 - record_run_statistics, 732
 - record_walltime, 732
 - set_model, 733
 - StepSize, 726
 - Success, 726
 - total_backtracks, 735
 - total_der_evals, 736
 - total_fun_evals, 736
 - total_iterations, 736
 - total_walltime, 736
 - TrustRegionRadius, 726

- Unassigned, 726
- mappel::QuasiNewtonMaximizer< Model >, 724
- mappel::SimulatedAnnealingMaximizer
 - anneal, 739
 - clear_stats, 739
 - compute_estimate, 739, 740
 - compute_estimate_debug, 740
 - compute_profile_estimate, 740
 - cooling_rate, 744
 - Error, 739
 - estimate_max, 740, 741
 - estimate_max_debug, 741
 - estimate_max_stack, 741, 742
 - estimate_profile_stack, 742
 - exit_counts, 744
 - ExitCode, 739
 - FunctionChange, 739
 - get_debug_stats, 742
 - get_model, 742
 - get_stats, 742
 - GradRatio, 739
 - max_threads, 744
 - MaxBacktracks, 739
 - MaxIter, 739
 - model, 744
 - mtx, 744
 - name, 743
 - num_estimations, 745
 - num_iterations, 745
 - num_threads, 745
 - NumExitCodes, 745
 - record_exit_code, 743
 - record_walltime, 743
 - set_model, 743
 - SimulatedAnnealingMaximizer, 739
 - StepSize, 739
 - Success, 739
 - T_init, 745
 - total_walltime, 745
 - TrustRegionRadius, 739
 - Unassigned, 739
- mappel::SimulatedAnnealingMaximizer< Model >, 737
- mappel::ThreadedEstimator
 - clear_stats, 771
 - compute_estimate, 771
 - compute_estimate_debug, 772
 - compute_profile_estimate, 772
 - Error, 771
 - estimate_max, 772, 773
 - estimate_max_debug, 773
 - estimate_max_stack, 773, 774
 - estimate_profile_stack, 774
 - exit_counts, 776
 - ExitCode, 771
 - FunctionChange, 771
 - get_debug_stats, 774
 - get_model, 774
 - get_stats, 774
 - GradRatio, 771
 - max_threads, 776
 - MaxBacktracks, 771
 - MaxIter, 771
 - model, 776
 - mtx, 776
 - name, 775
 - num_estimations, 776
 - num_threads, 777
 - NumExitCodes, 777
 - record_exit_code, 775
 - record_walltime, 775
 - set_model, 775
 - StepSize, 771
 - Success, 771
 - ThreadedEstimator, 771
 - total_walltime, 777
 - TrustRegionRadius, 771
 - Unassigned, 771
- mappel::ThreadedEstimator< Model >, 769
- mappel::TrustRegionMaximizer
 - alpha, 788
 - backtrack, 781
 - bound_step, 781
 - boundary_stepback_min_kappa, 788
 - clear_stats, 781
 - compute_D_scale, 782
 - compute_bound_scaling_vec, 781
 - compute_cauchy_point, 782
 - compute_estimate, 782
 - compute_estimate_debug, 782
 - compute_initial_trust_radius, 783
 - compute_profile_estimate, 783
 - convergence_test, 783
 - delta, 789
 - delta_decrease, 789
 - delta_decrease_min, 789
 - delta_increase, 789
 - delta_init_max, 789
 - delta_init_min, 789
 - epsilon, 789
 - Error, 780
 - estimate_max, 783, 784
 - estimate_max_debug, 784
 - estimate_max_stack, 785
 - estimate_profile_stack, 785
 - exit_counts, 789
 - ExitCode, 780
 - FunctionChange, 780
 - get_debug_stats, 785

- get_model, 785
- get_stats, 786
- GradRatio, 780
- lambda_min, 790
- last_backtrack_idx, 790
- local_maximize, 786
- max_backtracks, 790
- max_iterations, 790
- max_scaling, 790
- max_threads, 790
- MaxBacktracks, 780
- MaxIter, 780
- maximize, 786
- MaximizerData, 780
- mean_backtracks, 786
- mean_der_evals, 787
- mean_fun_evals, 787
- mean_iterations, 787
- min_scaling, 790
- model, 791
- mtx, 791
- name, 787
- num_estimations, 791
- num_threads, 791
- NumExitCodes, 791
- quadratic_model_value, 787
- record_exit_code, 787
- record_run_statistics, 787
- record_walltime, 787
- rho_cauchy_min, 792
- rho_obj_min, 792
- rho_obj_opt, 792
- set_model, 788
- solve_TR_subproblem, 788
- solve_restricted_step_length_newton, 788
- StepSize, 780
- Success, 780
- total_backtracks, 792
- total_der_evals, 792
- total_fun_evals, 792
- total_iterations, 792
- total_walltime, 792
- TrustRegionMaximizer, 781
- TrustRegionRadius, 780
- Unassigned, 780
- mappel::TrustRegionMaximizer< Model >, 777
- mappel::mcmc, 49
 - estimate_sample_posterior, 49
 - num_oversample, 49
 - sample_posterior, 49
 - sample_posterior_debug, 50
 - thin_sample, 50
- mappel::methods, 50
 - aposteriori_objective, 53
 - cr_lower_bound, 54
 - error_bounds_expected, 54
 - error_bounds_observed, 54
 - error_bounds_posterior_credible, 54
 - estimate_max, 54, 55
 - estimate_mcmc_posterior, 55
 - estimate_mcmc_sample, 56
 - expected_information, 56, 57
 - likelihood_objective, 57
 - make_estimator, 57
 - model_image, 58
 - observed_information, 58
 - prior_objective, 59
 - simulate_image, 59, 60
 - simulate_image_from_model, 60, 61
- mappel::methods::debug, 61
 - estimate_max_debug, 62
 - estimate_mcmc_sample_debug, 62
- mappel::methods::likelihood, 62
 - grad, 63
 - grad2, 63, 64
 - hessian, 64
 - llh, 64
 - rlh, 64, 65
- mappel::methods::likelihood::debug, 65
 - grad_components, 66
 - hessian_components, 66
 - llh_components, 66
 - rlh_components, 66, 67
- mappel::methods::objective, 67
 - grad, 68, 69
 - grad2, 69
 - hessian, 69, 70
 - llh, 70, 71
 - negative_definite_hessian, 71
 - rlh, 71, 72
- mappel::methods::objective::debug, 72
 - grad_components, 73
 - hessian_components, 73
 - llh_components, 74
 - rlh_components, 74
- mappel::methods::objective::openmp, 75
 - grad_stack, 75
 - hessian_stack, 76
 - llh_stack, 76, 77
 - negative_definite_hessian_stack, 77
 - rlh_stack, 78, 79
- mappel::methods::openmp, 79
 - cr_lower_bound_stack, 80
 - error_bounds_expected_stack, 80
 - error_bounds_observed_stack, 81
 - error_bounds_posterior_credible_stack, 81
 - estimate_max_stack, 81
 - estimate_mcmc_posterior_stack, 82

- [estimate_mcmc_sample_stack](#), [82](#)
 - [estimate_profile_likelihood](#), [82](#), [83](#)
 - [expected_information_stack](#), [83](#)
 - [model_image_stack](#), [83](#)
 - [sample_prior_stack](#), [84](#)
 - [simulate_image_stack](#), [84](#)
- [MappelError](#)
 - [mappel](#), [26](#)
- [MatT](#)
 - [mappel](#), [26](#)
- [max_backtracks](#)
 - [mappel::IterativeMaximizer](#), [542](#)
 - [mappel::NewtonDiagonalMaximizer](#), [669](#)
 - [mappel::NewtonMaximizer](#), [682](#)
 - [mappel::QuasiNewtonMaximizer](#), [734](#)
 - [mappel::TrustRegionMaximizer](#), [790](#)
- [max_iterations](#)
 - [mappel::IterativeMaximizer](#), [542](#)
 - [mappel::NewtonDiagonalMaximizer](#), [669](#)
 - [mappel::NewtonMaximizer](#), [682](#)
 - [mappel::QuasiNewtonMaximizer](#), [734](#)
 - [mappel::TrustRegionMaximizer](#), [790](#)
- [max_scaling](#)
 - [mappel::TrustRegionMaximizer](#), [790](#)
- [max_seq_len](#)
 - [mappel::IterativeMaximizer::MaximizerData](#), [565](#)
- [max_threads](#)
 - [mappel::CGaussHeuristicEstimator](#), [94](#)
 - [mappel::CGaussMLE](#), [102](#)
 - [mappel::HeuristicEstimator](#), [518](#)
 - [mappel::IterativeMaximizer](#), [542](#)
 - [mappel::NewtonDiagonalMaximizer](#), [669](#)
 - [mappel::NewtonMaximizer](#), [682](#)
 - [mappel::QuasiNewtonMaximizer](#), [734](#)
 - [mappel::SimulatedAnnealingMaximizer](#), [744](#)
 - [mappel::ThreadedEstimator](#), [776](#)
 - [mappel::TrustRegionMaximizer](#), [790](#)
- [MaxBacktracks](#)
 - [mappel::CGaussHeuristicEstimator](#), [89](#)
 - [mappel::CGaussMLE](#), [97](#)
 - [mappel::Estimator](#), [105](#)
 - [mappel::HeuristicEstimator](#), [513](#)
 - [mappel::IterativeMaximizer](#), [535](#)
 - [mappel::NewtonDiagonalMaximizer](#), [661](#)
 - [mappel::NewtonMaximizer](#), [674](#)
 - [mappel::QuasiNewtonMaximizer](#), [726](#)
 - [mappel::SimulatedAnnealingMaximizer](#), [739](#)
 - [mappel::ThreadedEstimator](#), [771](#)
 - [mappel::TrustRegionMaximizer](#), [780](#)
- [MaxIter](#)
 - [mappel::CGaussHeuristicEstimator](#), [89](#)
 - [mappel::CGaussMLE](#), [97](#)
 - [mappel::Estimator](#), [105](#)
 - [mappel::HeuristicEstimator](#), [513](#)
 - [mappel::IterativeMaximizer](#), [535](#)
 - [mappel::NewtonDiagonalMaximizer](#), [661](#)
 - [mappel::NewtonMaximizer](#), [674](#)
 - [mappel::QuasiNewtonMaximizer](#), [726](#)
 - [mappel::SimulatedAnnealingMaximizer](#), [739](#)
 - [mappel::ThreadedEstimator](#), [771](#)
 - [mappel::TrustRegionMaximizer](#), [780](#)
- [maxidx](#)
 - [mappel](#), [39](#)
- [maximize](#)
 - [mappel::IterativeMaximizer](#), [540](#)
 - [mappel::NewtonDiagonalMaximizer](#), [666](#)
 - [mappel::NewtonMaximizer](#), [679](#)
 - [mappel::QuasiNewtonMaximizer](#), [731](#)
 - [mappel::TrustRegionMaximizer](#), [786](#)
- [MaximizerData](#)
 - [mappel::IterativeMaximizer::MaximizerData](#), [561](#)
 - [mappel::NewtonDiagonalMaximizer](#), [661](#)
 - [mappel::NewtonMaximizer](#), [674](#)
 - [mappel::QuasiNewtonMaximizer](#), [726](#)
 - [mappel::TrustRegionMaximizer](#), [780](#)
- [mcmc.cpp](#), [818](#)
- [mcmc.h](#), [819](#)
- [mcmc_candidate_eta_sigma](#)
 - [mappel::Gauss2DsxyModel](#), [509](#)
- [mcmc_candidate_eta_y](#)
 - [mappel::Gauss2DsxyModel](#), [509](#)
- [mean_backtracks](#)
 - [mappel::IterativeMaximizer](#), [540](#)
 - [mappel::NewtonDiagonalMaximizer](#), [666](#)
 - [mappel::NewtonMaximizer](#), [679](#)
 - [mappel::QuasiNewtonMaximizer](#), [731](#)
 - [mappel::TrustRegionMaximizer](#), [786](#)
- [mean_der_evals](#)
 - [mappel::IterativeMaximizer](#), [540](#)
 - [mappel::NewtonDiagonalMaximizer](#), [667](#)
 - [mappel::NewtonMaximizer](#), [680](#)
 - [mappel::QuasiNewtonMaximizer](#), [732](#)
 - [mappel::TrustRegionMaximizer](#), [787](#)
- [mean_fun_evals](#)
 - [mappel::IterativeMaximizer](#), [540](#)
 - [mappel::NewtonDiagonalMaximizer](#), [667](#)
 - [mappel::NewtonMaximizer](#), [680](#)
 - [mappel::QuasiNewtonMaximizer](#), [732](#)
 - [mappel::TrustRegionMaximizer](#), [787](#)
- [mean_iterations](#)
 - [mappel::IterativeMaximizer](#), [540](#)
 - [mappel::NewtonDiagonalMaximizer](#), [667](#)
 - [mappel::NewtonMaximizer](#), [680](#)
 - [mappel::QuasiNewtonMaximizer](#), [732](#)
 - [mappel::TrustRegionMaximizer](#), [787](#)
- [min_scaling](#)
 - [mappel::TrustRegionMaximizer](#), [790](#)
- [min_sigma](#)

- mappel::Gauss2DsMAP, 395
- mappel::Gauss2DsMLE, 427
- mappel::Gauss2DsModel, 457
- mappel::Gauss2DsxyModel, 509
- model
 - mappel::CGaussHeuristicEstimator, 94
 - mappel::CGaussMLE, 102
 - mappel::Estimator, 110
 - mappel::Gauss1DModel::Stencil, 749
 - mappel::Gauss1DsModel::Stencil, 753
 - mappel::Gauss2DModel::Stencil, 768
 - mappel::Gauss2DsModel::Stencil, 763
 - mappel::Gauss2DsxyModel::Stencil, 757
 - mappel::HeuristicEstimator, 518
 - mappel::IterativeMaximizer, 542
 - mappel::NewtonDiagonalMaximizer, 669
 - mappel::NewtonMaximizer, 682
 - mappel::QuasiNewtonMaximizer, 734
 - mappel::SimulatedAnnealingMaximizer, 744
 - mappel::ThreadedEstimator, 776
 - mappel::TrustRegionMaximizer, 791
- model_grad
 - PoissonGaussianNoise2DObjective.h, 843
- model_grad2
 - PoissonGaussianNoise2DObjective.h, 843
- model_grad_stack
 - mappel, 39
- model_hessian
 - PoissonGaussianNoise2DObjective.h, 843
- model_hessian_stack
 - mappel, 40
- model_image
 - mappel::methods, 58
 - PoissonGaussianNoise2DObjective.h, 843
- model_image_stack
 - mappel, 40
 - mappel::methods::openmp, 83
- model_methods.h, 827
- model_methods_impl.h, 830
- model_positive_hessian_stack
 - mappel, 41
- ModelBoundsError
 - mappel::ModelBoundsError, 658
- ModelDataStackT
 - mappel, 26
 - mappel::Gauss1DMAP, 117
 - mappel::Gauss1DMLE, 144
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 228
 - mappel::Gauss2DMAP, 283
 - mappel::Gauss2DMLE, 313
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsxyMAP, 466
 - mappel::PoissonNoise1DObjective, 709
 - mappel::PoissonNoise2DObjective, 717
 - PoissonGaussianNoise2DObjective, 705
- ModelDataT
 - mappel, 26
 - mappel::Gauss1DMAP, 117
 - mappel::Gauss1DMLE, 145
 - mappel::Gauss1DsMAP, 200
 - mappel::Gauss1DsMLE, 228
 - mappel::Gauss2DMAP, 283
 - mappel::Gauss2DMLE, 313
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsxyMAP, 467
 - mappel::PoissonNoise1DObjective, 709
 - mappel::PoissonNoise2DObjective, 717
 - PoissonGaussianNoise2DObjective, 705
- modified_cholesky
 - mappel, 41
- mtx
 - mappel::CGaussHeuristicEstimator, 94
 - mappel::CGaussMLE, 102
 - mappel::HeuristicEstimator, 518
 - mappel::IterativeMaximizer, 543
 - mappel::NewtonDiagonalMaximizer, 670
 - mappel::NewtonMaximizer, 683
 - mappel::QuasiNewtonMaximizer, 735
 - mappel::SimulatedAnnealingMaximizer, 744
 - mappel::ThreadedEstimator, 776
 - mappel::TrustRegionMaximizer, 791
- nBacktracks
 - mappel::IterativeMaximizer::MaximizerData, 566
- nIterations
 - mappel::IterativeMaximizer::MaximizerData, 566
- name
 - mappel::CGaussHeuristicEstimator, 93
 - mappel::CGaussMLE, 101
 - mappel::Estimator, 109
 - mappel::Gauss1DMAP, 136
 - mappel::Gauss1DMLE, 164
 - mappel::Gauss1DsMAP, 220
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss2DMAP, 304
 - mappel::Gauss2DMLE, 334
 - mappel::Gauss2DsMAP, 395
 - mappel::Gauss2DsMLE, 427
 - mappel::Gauss2DsxyMAP, 485
 - mappel::HeuristicEstimator, 517
 - mappel::IterativeMaximizer, 540
 - mappel::NewtonDiagonalMaximizer, 667
 - mappel::NewtonMaximizer, 680
 - mappel::QuasiNewtonMaximizer, 732
 - mappel::SimulatedAnnealingMaximizer, 743

- mappel::ThreadedEstimator, 775
- mappel::TrustRegionMaximizer, 787
- negative_definite_hessian
 - mappel::methods::objective, 71
- negative_definite_hessian_stack
 - mappel::methods::objective::openmp, 77
- NewtonDiagonalMaximizer
 - mappel::NewtonDiagonalMaximizer, 661
- NewtonMaximizer
 - mappel::NewtonMaximizer, 674
- normal_prior_grad
 - mappel, 42
- normal_prior_grad2
 - mappel, 42
- normal_quantile_onesided
 - mappel, 42
- normal_quantile_twosided
 - mappel, 42
- NotImplementedError
 - mappel::NotImplementedError, 685
- num_dim
 - mappel::Gauss1DMap, 136
 - mappel::Gauss1DMLE, 164
 - mappel::Gauss1DModel, 191
 - mappel::Gauss1DsMAP, 220
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 275
 - mappel::Gauss2DMap, 304
 - mappel::Gauss2DMLE, 334
 - mappel::Gauss2DModel, 363
 - mappel::Gauss2DsMAP, 395
 - mappel::Gauss2DsMLE, 427
 - mappel::Gauss2DsModel, 458
 - mappel::Gauss2DsxyMAP, 486
 - mappel::Gauss2DsxyModel, 509
 - mappel::ImageFormat1DBase, 525
 - mappel::ImageFormat2DBase, 531
 - mappel::PoissonNoise1DObjective, 713
 - mappel::PoissonNoise2DObjective, 720
- num_estimations
 - mappel::CGaussHeuristicEstimator, 94
 - mappel::CGaussMLE, 103
 - mappel::Estimator, 111
 - mappel::HeuristicEstimator, 518
 - mappel::IterativeMaximizer, 543
 - mappel::NewtonDiagonalMaximizer, 670
 - mappel::NewtonMaximizer, 683
 - mappel::QuasiNewtonMaximizer, 735
 - mappel::SimulatedAnnealingMaximizer, 745
 - mappel::ThreadedEstimator, 776
 - mappel::TrustRegionMaximizer, 791
- num_fixed_parameters
 - mappel::IterativeMaximizer::MaximizerData, 566
- num_hyperparams
 - mappel::Gauss1DMap, 137
 - mappel::Gauss1DMLE, 165
 - mappel::Gauss1DModel, 192
 - mappel::Gauss1DsMAP, 220
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 275
 - mappel::Gauss2DMap, 304
 - mappel::Gauss2DMLE, 334
 - mappel::Gauss2DModel, 363
 - mappel::Gauss2DsMAP, 395
 - mappel::Gauss2DsMLE, 427
 - mappel::Gauss2DsModel, 458
 - mappel::Gauss2DsxyMAP, 486
 - mappel::Gauss2DsxyModel, 509
 - mappel::MAPEstimator, 559
 - mappel::MCMCAdaptor1Ds, 602
 - mappel::MCMCAdaptor1D, 584
 - mappel::MCMCAdaptor2Ds, 639
 - mappel::MCMCAdaptor2D, 620
 - mappel::MLEstimator, 657
 - mappel::PointEmitterModel, 703
- num_iterations
 - mappel::CGaussMLE, 103
 - mappel::SimulatedAnnealingMaximizer, 745
- num_oversample
 - mappel::mcmc, 49
- num_params
 - mappel::Gauss1DMap, 137
 - mappel::Gauss1DMLE, 165
 - mappel::Gauss1DModel, 192
 - mappel::Gauss1DsMAP, 220
 - mappel::Gauss1DsMLE, 248
 - mappel::Gauss1DsModel, 275
 - mappel::Gauss2DMap, 304
 - mappel::Gauss2DMLE, 334
 - mappel::Gauss2DModel, 363
 - mappel::Gauss2DsMAP, 395
 - mappel::Gauss2DsMLE, 427
 - mappel::Gauss2DsModel, 458
 - mappel::Gauss2DsxyMAP, 486
 - mappel::Gauss2DsxyModel, 509
 - mappel::MAPEstimator, 559
 - mappel::MCMCAdaptor1Ds, 602
 - mappel::MCMCAdaptor1D, 584
 - mappel::MCMCAdaptor2Ds, 639
 - mappel::MCMCAdaptor2D, 620
 - mappel::MLEstimator, 657
 - mappel::PointEmitterModel, 703
- num_phases
 - mappel::Gauss1DMap, 137
 - mappel::Gauss1DMLE, 165
 - mappel::Gauss1DModel, 192
 - mappel::Gauss1DsMAP, 220
 - mappel::Gauss1DsMLE, 249

- mappel::Gauss1DsModel, 275
- mappel::Gauss2DModel, 304
- mappel::Gauss2DModel, 334
- mappel::Gauss2DsMAP, 363
- mappel::Gauss2DsMAP, 395
- mappel::Gauss2DsMLE, 427
- mappel::Gauss2DsModel, 458
- mappel::Gauss2DsxyMAP, 486
- mappel::MCMCAdaptor1Ds, 602
- mappel::MCMCAdaptor1D, 584
- mappel::MCMCAdaptor2Ds, 639
- mappel::MCMCAdaptor2D, 621
- mappel::MCMCAdaptorBase, 643
- num_threads
 - mappel::CGaussHeuristicEstimator, 94
 - mappel::CGaussMLE, 103
 - mappel::HeuristicEstimator, 519
 - mappel::IterativeMaximizer, 543
 - mappel::NewtonDiagonalMaximizer, 670
 - mappel::NewtonMaximizer, 683
 - mappel::QuasiNewtonMaximizer, 735
 - mappel::SimulatedAnnealingMaximizer, 745
 - mappel::ThreadedEstimator, 777
 - mappel::TrustRegionMaximizer, 791
- NumExitCodes
 - mappel::CGaussHeuristicEstimator, 95
 - mappel::CGaussMLE, 103
 - mappel::Estimator, 111
 - mappel::HeuristicEstimator, 519
 - mappel::IterativeMaximizer, 543
 - mappel::NewtonDiagonalMaximizer, 670
 - mappel::NewtonMaximizer, 683
 - mappel::QuasiNewtonMaximizer, 735
 - mappel::SimulatedAnnealingMaximizer, 745
 - mappel::ThreadedEstimator, 777
 - mappel::TrustRegionMaximizer, 791
- numerical.cpp, 833
- numerical.h, 834
- NumericalError
 - mappel::NumericalError, 686
- OMPEXceptionCatcher
 - omp_exception_catcher, 85
 - omp_exception_catcher::impl_::OMPEXceptionCatcher, 687
- OMPEXceptionCatcher.h, 835
- observed_information
 - mappel::methods, 58
- omp_exception_catcher, 85
 - OMPEXceptionCatcher, 85
- omp_exception_catcher::impl_, 85
- omp_exception_catcher::impl_::OMPEXceptionCatcher
 - Abort, 687
 - Continue, 687
 - DoNotTry, 687
 - OMPEXceptionCatcher, 687
 - rethrow, 687
 - RethrowFirst, 687
 - run, 687
 - setGlobalDefaultStrategy, 688
 - Strategy, 687
- omp_exception_catcher::impl_::OMPEXceptionCatcher<
 - IntType >, 686
- openmp_methods.h, 836
- operator<<
 - mappel, 42, 43
 - mappel::Estimator, 110
 - mappel::Gauss1DModel::Stencil, 748
 - mappel::Gauss1DsModel::Stencil, 752
 - mappel::Gauss2DModel::Stencil, 767
 - mappel::Gauss2DsModel::Stencil, 761
 - mappel::Gauss2DsxyModel::Stencil, 756
- operator=
 - mappel::Gauss1DModel, 127, 128
 - mappel::Gauss1DModel, 155
 - mappel::Gauss1DModel, 182, 183
 - mappel::Gauss1DsMAP, 210
 - mappel::Gauss1DsMLE, 238, 239
 - mappel::Gauss1DsModel, 266
 - mappel::Gauss2DModel, 295
 - mappel::Gauss2DModel, 325
 - mappel::Gauss2DModel, 354
 - mappel::Gauss2DsMAP, 384, 385
 - mappel::Gauss2DsMLE, 416, 417
 - mappel::Gauss2DsModel, 448
 - mappel::ImageFormat2DBase, 530, 531
 - mappel::MAPEstimator, 554
 - mappel::MCMCAdaptor1Ds, 595
 - mappel::MCMCAdaptor1D, 577
 - mappel::MCMCAdaptor2Ds, 631, 632
 - mappel::MCMCAdaptor2D, 613
 - mappel::MLEstimator, 652
 - mappel::PointEmitterModel, 698
 - mappel::PoissonNoise1DObjective, 712
 - mappel::PoissonNoise2DObjective, 719
- ParallelRngGeneratorT
 - mappel, 26
- ParallelRngManagerT
 - mappel, 26
- ParamMatT
 - mappel::PriorMAP1DObjective, 722
- ParamVecT
 - mappel, 26
 - mappel::Gauss1DModel, 117
 - mappel::Gauss1DModel, 145
 - mappel::Gauss1DModel, 173
 - mappel::Gauss1DsMAP, 200

- mappel::Gauss1DsMLE, 228
- mappel::Gauss1DsModel, 256
- mappel::Gauss2DMAP, 284
- mappel::Gauss2DMLE, 314
- mappel::Gauss2DModel, 343
- mappel::Gauss2DsMAP, 372
- mappel::Gauss2DsMLE, 404
- mappel::Gauss2DsModel, 436
- mappel::Gauss2DsxyMAP, 467
- mappel::Gauss2DsxyModel, 493
- mappel::MAPEstimator, 548
- mappel::MCMCAdaptor1Ds, 589
- mappel::MCMCAdaptor1D, 570
- mappel::MCMCAdaptor2Ds, 625
- mappel::MCMCAdaptor2D, 607
- mappel::MLEstimator, 646
- mappel::PointEmitterModel, 692
- ParameterValueError
 - mappel::ParameterValueError, 689
- ParamT
 - mappel, 26
 - mappel::Gauss1DMAP, 117
 - mappel::Gauss1DMLE, 145
 - mappel::Gauss1DModel, 173
 - mappel::Gauss1DModel::Stencil, 746
 - mappel::Gauss1DsMAP, 200
 - mappel::Gauss1DsMLE, 228
 - mappel::Gauss1DsModel, 256
 - mappel::Gauss1DsModel::Stencil, 750
 - mappel::Gauss2DMAP, 283
 - mappel::Gauss2DMLE, 313
 - mappel::Gauss2DModel, 343
 - mappel::Gauss2DModel::Stencil, 765
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsModel::Stencil, 759
 - mappel::Gauss2DsxyMAP, 467
 - mappel::Gauss2DsxyModel, 493
 - mappel::Gauss2DsxyModel::Stencil, 754
 - mappel::MAPEstimator, 548
 - mappel::MCMCAdaptor1Ds, 589
 - mappel::MCMCAdaptor1D, 570
 - mappel::MCMCAdaptor2Ds, 625
 - mappel::MCMCAdaptor2D, 607
 - mappel::MLEstimator, 646
 - mappel::PointEmitterModel, 692
 - mappel::PriorMAP1DObjective, 722
- pareto_prior_grad
 - mappel, 43
- pareto_prior_grad2
 - mappel, 43
- pixel_grad
 - mappel::Gauss1DMAP, 128
- mappel::Gauss1DMLE, 156
- mappel::Gauss1DModel, 183
- mappel::Gauss1DsMAP, 211
- mappel::Gauss1DsMLE, 239
- mappel::Gauss1DsModel, 266
- mappel::Gauss2DMAP, 295
- mappel::Gauss2DMLE, 325
- mappel::Gauss2DModel, 354
- mappel::Gauss2DsMAP, 385
- mappel::Gauss2DsMLE, 417
- mappel::Gauss2DsModel, 448
- mappel::Gauss2DsxyMAP, 477
- mappel::Gauss2DsxyModel, 503
- pixel_grad2
 - mappel::Gauss1DMAP, 128
 - mappel::Gauss1DMLE, 156
 - mappel::Gauss1DModel, 183
 - mappel::Gauss1DsMAP, 211
 - mappel::Gauss1DsMLE, 239
 - mappel::Gauss1DsModel, 266
 - mappel::Gauss2DMAP, 295
 - mappel::Gauss2DMLE, 325
 - mappel::Gauss2DModel, 354
 - mappel::Gauss2DsMAP, 385
 - mappel::Gauss2DsMLE, 417
 - mappel::Gauss2DsModel, 448
 - mappel::Gauss2DsxyMAP, 477
 - mappel::Gauss2DsxyModel, 503
- pixel_hess
 - mappel::Gauss1DMAP, 128
 - mappel::Gauss1DMLE, 156
 - mappel::Gauss1DModel, 183
 - mappel::Gauss1DsMAP, 211
 - mappel::Gauss1DsMLE, 239
 - mappel::Gauss1DsModel, 266
 - mappel::Gauss2DMAP, 295
 - mappel::Gauss2DMLE, 325
 - mappel::Gauss2DModel, 354
 - mappel::Gauss2DsMAP, 385
 - mappel::Gauss2DsMLE, 417
 - mappel::Gauss2DsModel, 449
 - mappel::Gauss2DsxyMAP, 477
 - mappel::Gauss2DsxyModel, 503
- pixel_hess_update
 - mappel::Gauss1DMAP, 128
 - mappel::Gauss1DMLE, 156
 - mappel::Gauss1DModel, 183
 - mappel::Gauss1DsMAP, 211
 - mappel::Gauss1DsMLE, 239
 - mappel::Gauss1DsModel, 267
 - mappel::Gauss2DMAP, 296
 - mappel::Gauss2DMLE, 326
 - mappel::Gauss2DModel, 355
 - mappel::Gauss2DsMAP, 385

- mappel::Gauss2DsMLE, 417
- mappel::Gauss2DsModel, 449
- mappel::Gauss2DsxyMAP, 477
- mappel::Gauss2DsxyModel, 503
- pixel_model_value
 - mappel::Gauss1DMAP, 128
 - mappel::Gauss1DMLE, 156
 - mappel::Gauss1DModel, 183
 - mappel::Gauss1DsMAP, 211
 - mappel::Gauss1DsMLE, 239
 - mappel::Gauss1DsModel, 267
 - mappel::Gauss2DMAP, 296
 - mappel::Gauss2DMLE, 326
 - mappel::Gauss2DModel, 355
 - mappel::Gauss2DsMAP, 386
 - mappel::Gauss2DsMLE, 418
 - mappel::Gauss2DsModel, 449
 - mappel::Gauss2DsxyMAP, 477
 - mappel::Gauss2DsxyModel, 503
- PointEmitterModel
 - mappel::PointEmitterModel, 692, 693
- PointEmitterModel.cpp, 839
- PointEmitterModel.h, 840
- poisson_log_likelihood
 - mappel, 43
- PoissonGaussianNoise2DObjective
 - CoordIdxT, 705
 - CoordStackT, 705
 - CoordT, 705
 - estimator_names, 706
 - ModelDataStackT, 705
 - ModelDataT, 705
 - PoissonGaussianNoise2DObjective, 706
 - sensor_bg_map, 706
 - sensor_gain_map, 706
- PoissonGaussianNoise2DObjective< ModelBase >, 704
- PoissonGaussianNoise2DObjective.cpp, 841
- PoissonGaussianNoise2DObjective.h, 841
 - fisher_information, 842
 - log_likelihood, 842
 - make_estimator, 842
 - model_grad, 843
 - model_grad2, 843
 - model_hessian, 843
 - model_image, 843
 - relative_log_likelihood, 843
 - simulate_image, 843, 844
- PoissonNoise1DObjective
 - mappel::PoissonNoise1DObjective, 710
- PoissonNoise1DObjective.cpp, 844
- PoissonNoise1DObjective.h, 845
- PoissonNoise2DObjective
 - mappel::PoissonNoise2DObjective, 717
- PoissonNoise2DObjective.cpp, 846
- PoissonNoise2DObjective.h, 847
- print_centered_title
 - mappel, 43
- print_image
 - mappel, 44
- print_labeled_image
 - mappel, 44
- print_vec_row
 - mappel, 44
- prior
 - mappel::Gauss1DMAP, 137
 - mappel::Gauss1DMLE, 165
 - mappel::Gauss1DModel, 192
 - mappel::Gauss1DsMAP, 221
 - mappel::Gauss1DsMLE, 249
 - mappel::Gauss1DsModel, 276
 - mappel::Gauss2DMAP, 305
 - mappel::Gauss2DMLE, 335
 - mappel::Gauss2DModel, 363
 - mappel::Gauss2DsMAP, 396
 - mappel::Gauss2DsMLE, 428
 - mappel::Gauss2DsModel, 458
 - mappel::Gauss2DsxyMAP, 486
 - mappel::Gauss2DsxyModel, 510
 - mappel::MAPEstimator, 559
 - mappel::MCMCAdaptor1Ds, 603
 - mappel::MCMCAdaptor1D, 585
 - mappel::MCMCAdaptor2Ds, 639
 - mappel::MCMCAdaptor2D, 621
 - mappel::MLEstimator, 657
 - mappel::PointEmitterModel, 703
- prior_grad2_update
 - mappel::PriorMAP1DObjective, 722
- prior_grad_update
 - mappel::PriorMAP1DObjective, 722
- prior_hess_update
 - mappel::PriorMAP1DObjective, 722
- prior_log_likelihood
 - mappel::PriorMAP1DObjective, 722
- prior_objective
 - mappel::methods, 59
- prior_relative_log_likelihood
 - mappel::PriorMAP1DObjective, 723
- prior_types
 - mappel::Gauss1DMAP, 137
 - mappel::Gauss1DMLE, 165
 - mappel::Gauss1DModel, 192
 - mappel::Gauss1DsMAP, 221
 - mappel::Gauss1DsMLE, 249
 - mappel::Gauss1DsModel, 276
 - mappel::Gauss2DMAP, 305
 - mappel::Gauss2DMLE, 335
 - mappel::Gauss2DModel, 364
 - mappel::Gauss2DsMAP, 396

- mappel::Gauss2DsMLE, 428
- mappel::Gauss2DsModel, 459
- mappel::Gauss2DsxyMAP, 486
- PriorMAP1DObjective.h, 849
- psf_sigma
 - mappel::Gauss1DMAP, 138
 - mappel::Gauss1DMLE, 166
 - mappel::Gauss1DModel, 193
 - mappel::Gauss2DMAP, 305
 - mappel::Gauss2DMLE, 335
 - mappel::Gauss2DModel, 364
 - mappel::Gauss2DsxyMAP, 487
- quadratic_model_value
 - mappel::TrustRegionMaximizer, 787
- QuasiNewtonMaximizer
 - mappel::QuasiNewtonMaximizer, 727
- README.md, 849
- record_backtrack
 - mappel::IterativeMaximizer::MaximizerData, 562
- record_exit_code
 - mappel::CGaussHeuristicEstimator, 93
 - mappel::CGaussMLE, 101
 - mappel::Estimator, 110
 - mappel::HeuristicEstimator, 517
 - mappel::IterativeMaximizer, 540
 - mappel::NewtonDiagonalMaximizer, 667
 - mappel::NewtonMaximizer, 680
 - mappel::QuasiNewtonMaximizer, 732
 - mappel::SimulatedAnnealingMaximizer, 743
 - mappel::ThreadedEstimator, 775
 - mappel::TrustRegionMaximizer, 787
- record_iteration
 - mappel::IterativeMaximizer::MaximizerData, 562, 563
- record_run_statistics
 - mappel::IterativeMaximizer, 540
 - mappel::NewtonDiagonalMaximizer, 667
 - mappel::NewtonMaximizer, 680
 - mappel::QuasiNewtonMaximizer, 732
 - mappel::TrustRegionMaximizer, 787
- record_walltime
 - mappel::CGaussHeuristicEstimator, 93
 - mappel::CGaussMLE, 101
 - mappel::Estimator, 110
 - mappel::HeuristicEstimator, 517
 - mappel::IterativeMaximizer, 541
 - mappel::NewtonDiagonalMaximizer, 667
 - mappel::NewtonMaximizer, 680
 - mappel::QuasiNewtonMaximizer, 732
 - mappel::SimulatedAnnealingMaximizer, 743
 - mappel::ThreadedEstimator, 775
 - mappel::TrustRegionMaximizer, 787
- refine_gaussian_2Dmax
 - mappel, 44
- refine_gaussian_3Dmax
 - mappel, 44
- reflected_theta
 - mappel::Gauss1DMAP, 129
 - mappel::Gauss1DMLE, 156
 - mappel::Gauss1DModel, 183
 - mappel::Gauss1DsMAP, 211
 - mappel::Gauss1DsMLE, 240
 - mappel::Gauss1DsModel, 267
 - mappel::Gauss2DMAP, 296
 - mappel::Gauss2DMLE, 326
 - mappel::Gauss2DModel, 355
 - mappel::Gauss2DsMAP, 386
 - mappel::Gauss2DsMLE, 418
 - mappel::Gauss2DsModel, 449
 - mappel::Gauss2DsxyMAP, 477
 - mappel::Gauss2DsxyModel, 503
 - mappel::MAPEstimator, 554
 - mappel::MCMCAdaptor1Ds, 595
 - mappel::MCMCAdaptor1D, 577
 - mappel::MCMCAdaptor2Ds, 632
 - mappel::MCMCAdaptor2D, 613
 - mappel::MLEstimator, 652
 - mappel::PointEmitterModel, 698
- reflected_theta_stack
 - mappel::Gauss1DMAP, 129
 - mappel::Gauss1DMLE, 157
 - mappel::Gauss1DModel, 184
 - mappel::Gauss1DsMAP, 212
 - mappel::Gauss1DsMLE, 240
 - mappel::Gauss1DsModel, 267
 - mappel::Gauss2DMAP, 296
 - mappel::Gauss2DMLE, 326
 - mappel::Gauss2DModel, 355
 - mappel::Gauss2DsMAP, 386
 - mappel::Gauss2DsMLE, 418
 - mappel::Gauss2DsModel, 449
 - mappel::Gauss2DsxyMAP, 478
 - mappel::Gauss2DsxyModel, 503
 - mappel::MAPEstimator, 554
 - mappel::MCMCAdaptor1Ds, 596
 - mappel::MCMCAdaptor1D, 578
 - mappel::MCMCAdaptor2Ds, 632
 - mappel::MCMCAdaptor2D, 614
 - mappel::MLEstimator, 652
 - mappel::PointEmitterModel, 699
- relative_log_likelihood
 - PoissonGaussianNoise2DObjective.h, 843
- relative_poisson_log_likelihood
 - mappel, 45
- rename_hyperparam
 - mappel::Gauss1DMAP, 129
 - mappel::Gauss1DMLE, 157

- mappel::Gauss1DModel, [184](#)
- mappel::Gauss1DsMAP, [212](#)
- mappel::Gauss1DsMLE, [240](#)
- mappel::Gauss1DsModel, [267](#)
- mappel::Gauss2DMAP, [296](#)
- mappel::Gauss2DMLE, [326](#)
- mappel::Gauss2DModel, [355](#)
- mappel::Gauss2DsMAP, [386](#)
- mappel::Gauss2DsMLE, [418](#)
- mappel::Gauss2DsModel, [450](#)
- mappel::Gauss2DsxyMAP, [478](#)
- mappel::Gauss2DsxyModel, [504](#)
- mappel::MAPEstimator, [554](#)
- mappel::MCMCAdaptor1Ds, [596](#)
- mappel::MCMCAdaptor1D, [578](#)
- mappel::MCMCAdaptor2Ds, [632](#)
- mappel::MCMCAdaptor2D, [614](#)
- mappel::MLEstimator, [652](#)
- mappel::PointEmitterModel, [699](#)
- restore_stencil
 - mappel::IterativeMaximizer::MaximizerData, [563](#)
- restrict_value_range
 - mappel, [45](#)
- rethrow
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [687](#)
- RethrowFirst
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [687](#)
- ReturnIfSubclassT
 - mappel, [27](#)
- rho_cauchy_min
 - mappel::TrustRegionMaximizer, [792](#)
- rho_obj_min
 - mappel::TrustRegionMaximizer, [792](#)
- rho_obj_opt
 - mappel::TrustRegionMaximizer, [792](#)
- rllh
 - mappel::IterativeMaximizer::MaximizerData, [566](#)
 - mappel::methods::likelihood, [64](#), [65](#)
 - mappel::methods::objective, [71](#), [72](#)
- rllh_beta2_prior
 - mappel, [45](#)
- rllh_beta_prior
 - mappel, [45](#)
- rllh_components
 - mappel::methods::likelihood::debug, [66](#), [67](#)
 - mappel::methods::objective::debug, [74](#)
- rllh_gamma_prior
 - mappel, [45](#)
- rllh_normal_prior
 - mappel, [45](#)
- rllh_pareto_prior
 - mappel, [45](#)
- rllh_stack
 - mappel::methods::objective::openmp, [78](#), [79](#)
- rng.cpp, [849](#)
- rng.h, [850](#)
- rng_manager
 - mappel, [47](#)
- RngSeedT
 - mappel, [27](#)
- run
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [687](#)
- s0
 - mappel::IterativeMaximizer::MaximizerData, [566](#)
- s1
 - mappel::IterativeMaximizer::MaximizerData, [566](#)
- sample_mcmc_candidate
 - mappel::Gauss1DMAP, [129](#)
 - mappel::Gauss1DMLE, [157](#)
 - mappel::Gauss1DModel, [184](#)
 - mappel::Gauss1DsMAP, [212](#)
 - mappel::Gauss1DsMLE, [240](#)
 - mappel::Gauss1DsModel, [267](#)
 - mappel::Gauss2DMAP, [296](#)
 - mappel::Gauss2DMLE, [326](#)
 - mappel::Gauss2DModel, [355](#)
 - mappel::Gauss2DsMAP, [386](#)
 - mappel::Gauss2DsMLE, [418](#)
 - mappel::Gauss2DsModel, [450](#)
 - mappel::Gauss2DsxyMAP, [478](#)
 - mappel::Gauss2DsxyModel, [504](#)
 - mappel::MCMCAdaptor1Ds, [596](#)
 - mappel::MCMCAdaptor1D, [578](#)
 - mappel::MCMCAdaptor2Ds, [632](#)
 - mappel::MCMCAdaptor2D, [614](#)
- sample_posterior
 - mappel::mcmc, [49](#)
- sample_posterior_debug
 - mappel::mcmc, [50](#)
- sample_prior
 - mappel::Gauss1DMAP, [129](#)
 - mappel::Gauss1DMLE, [157](#)
 - mappel::Gauss1DModel, [184](#)
 - mappel::Gauss1DsMAP, [212](#)
 - mappel::Gauss1DsMLE, [240](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DMAP, [297](#)
 - mappel::Gauss2DMLE, [327](#)
 - mappel::Gauss2DModel, [356](#)
 - mappel::Gauss2DsMAP, [387](#)
 - mappel::Gauss2DsMLE, [419](#)
 - mappel::Gauss2DsModel, [450](#)
 - mappel::Gauss2DsxyMAP, [478](#)
 - mappel::Gauss2DsxyModel, [504](#)

- mappel::MAPEstimator, [554](#), [555](#)
- mappel::MCMCAdaptor1Ds, [596](#)
- mappel::MCMCAdaptor1D, [578](#)
- mappel::MCMCAdaptor2Ds, [632](#), [633](#)
- mappel::MCMCAdaptor2D, [614](#)
- mappel::MLEstimator, [652](#), [653](#)
- mappel::PointEmitterModel, [699](#)
- sample_prior_stack
 - mappel, [45](#)
 - mappel::methods::openmp, [84](#)
- save_seq
 - mappel::IterativeMaximizer::MaximizerData, [566](#)
- save_stencil
 - mappel::IterativeMaximizer::MaximizerData, [563](#)
- saved_stencil
 - mappel::IterativeMaximizer::MaximizerData, [563](#)
- saved_theta
 - mappel::IterativeMaximizer::MaximizerData, [563](#)
- sensor_bg_map
 - PoissonGaussianNoise2DObjective, [706](#)
- sensor_gain_map
 - PoissonGaussianNoise2DObjective, [706](#)
- seq_len
 - mappel::IterativeMaximizer::MaximizerData, [567](#)
- seq_rllh
 - mappel::IterativeMaximizer::MaximizerData, [567](#)
- set_background_mcmc_sampling
 - mappel::Gauss1DMAP, [130](#)
 - mappel::Gauss1DMLE, [157](#)
 - mappel::Gauss1DModel, [184](#)
 - mappel::Gauss1DsMAP, [212](#)
 - mappel::Gauss1DsMLE, [241](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DMAP, [297](#)
 - mappel::Gauss2DMLE, [327](#)
 - mappel::Gauss2DModel, [356](#)
 - mappel::Gauss2DsMAP, [387](#)
 - mappel::Gauss2DsMLE, [419](#)
 - mappel::Gauss2DsModel, [450](#)
 - mappel::Gauss2DsxyMAP, [478](#)
 - mappel::MCMCAdaptor1Ds, [596](#)
 - mappel::MCMCAdaptor1D, [578](#)
 - mappel::MCMCAdaptor2Ds, [633](#)
 - mappel::MCMCAdaptor2D, [614](#)
- set_bounds
 - mappel::Gauss1DMAP, [130](#)
 - mappel::Gauss1DMLE, [158](#)
 - mappel::Gauss1DModel, [185](#)
 - mappel::Gauss1DsMAP, [213](#)
 - mappel::Gauss1DsMLE, [241](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DMAP, [297](#)
 - mappel::Gauss2DMLE, [327](#)
 - mappel::Gauss2DModel, [356](#)
- mappel::Gauss2DsMAP, [387](#)
- mappel::Gauss2DsMLE, [419](#)
- mappel::Gauss2DsModel, [450](#)
- mappel::Gauss2DsxyMAP, [479](#)
- mappel::Gauss2DsxyModel, [504](#)
- mappel::MAPEstimator, [555](#)
- mappel::MCMCAdaptor1Ds, [597](#)
- mappel::MCMCAdaptor1D, [579](#)
- mappel::MCMCAdaptor2Ds, [633](#)
- mappel::MCMCAdaptor2D, [615](#)
- mappel::MLEstimator, [653](#)
- mappel::PointEmitterModel, [699](#)
- set_fixed_parameters
 - mappel::IterativeMaximizer::MaximizerData, [564](#)
- set_hyperparam_names
 - mappel::Gauss1DMAP, [130](#)
 - mappel::Gauss1DMLE, [158](#)
 - mappel::Gauss1DModel, [185](#)
 - mappel::Gauss1DsMAP, [213](#)
 - mappel::Gauss1DsMLE, [241](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DMAP, [297](#)
 - mappel::Gauss2DMLE, [327](#)
 - mappel::Gauss2DModel, [356](#)
 - mappel::Gauss2DsMAP, [387](#)
 - mappel::Gauss2DsMLE, [419](#)
 - mappel::Gauss2DsModel, [451](#)
 - mappel::Gauss2DsxyMAP, [479](#)
 - mappel::Gauss2DsxyModel, [504](#)
 - mappel::MAPEstimator, [555](#)
 - mappel::MCMCAdaptor1Ds, [597](#)
 - mappel::MCMCAdaptor1D, [579](#)
 - mappel::MCMCAdaptor2Ds, [633](#)
 - mappel::MCMCAdaptor2D, [615](#)
 - mappel::MLEstimator, [653](#)
 - mappel::PointEmitterModel, [699](#)
- set_hyperparam_value
 - mappel::Gauss1DMAP, [130](#)
 - mappel::Gauss1DMLE, [158](#)
 - mappel::Gauss1DModel, [185](#)
 - mappel::Gauss1DsMAP, [213](#)
 - mappel::Gauss1DsMLE, [241](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DMAP, [297](#)
 - mappel::Gauss2DMLE, [327](#)
 - mappel::Gauss2DModel, [356](#)
 - mappel::Gauss2DsMAP, [387](#)
 - mappel::Gauss2DsMLE, [419](#)
 - mappel::Gauss2DsModel, [451](#)
 - mappel::Gauss2DsxyMAP, [479](#)
 - mappel::Gauss2DsxyModel, [504](#)
 - mappel::MAPEstimator, [555](#)
 - mappel::MCMCAdaptor1Ds, [597](#)
 - mappel::MCMCAdaptor1D, [579](#)

- mappel::MCMCAdaptor2Ds, 633
- mappel::MCMCAdaptor2D, 615
- mappel::MLEstimator, 653
- mappel::PointEmitterModel, 699
- set_hyperparameters
 - mappel::PriorMAP1DObjective, 723
- set_hyperparams
 - mappel::Gauss1DMap, 130
 - mappel::Gauss1DMLE, 158
 - mappel::Gauss1DModel, 185
 - mappel::Gauss1DsMAP, 213
 - mappel::Gauss1DsMLE, 241
 - mappel::Gauss1DsModel, 269
 - mappel::Gauss2DMap, 298
 - mappel::Gauss2DMLE, 328
 - mappel::Gauss2DModel, 357
 - mappel::Gauss2DsMAP, 388
 - mappel::Gauss2DsMLE, 420
 - mappel::Gauss2DsModel, 451
 - mappel::Gauss2DsxyMAP, 479
 - mappel::Gauss2DsxyModel, 505
 - mappel::MAPEstimator, 555
 - mappel::MCMCAdaptor1Ds, 597
 - mappel::MCMCAdaptor1D, 579
 - mappel::MCMCAdaptor2Ds, 633
 - mappel::MCMCAdaptor2D, 615
 - mappel::MLEstimator, 653
 - mappel::PointEmitterModel, 700
- set_image_in_stack
 - mappel::Gauss1DMap, 130
 - mappel::Gauss1DMLE, 158
 - mappel::Gauss1DModel, 185
 - mappel::Gauss1DsMAP, 213
 - mappel::Gauss1DsMLE, 241
 - mappel::Gauss1DsModel, 269
 - mappel::Gauss2DMap, 298
 - mappel::Gauss2DMLE, 328
 - mappel::Gauss2DModel, 357
 - mappel::Gauss2DsMAP, 388
 - mappel::Gauss2DsMLE, 420
 - mappel::Gauss2DsModel, 451
 - mappel::Gauss2DsxyMAP, 479
 - mappel::Gauss2DsxyModel, 505
 - mappel::ImageFormat1DBase, 524
 - mappel::ImageFormat2DBase, 531
 - mappel::PoissonNoise1DObjective, 712
 - mappel::PoissonNoise2DObjective, 719
- set_intensity_mcmc_sampling
 - mappel::Gauss1DMap, 131
 - mappel::Gauss1DMLE, 158
 - mappel::Gauss1DModel, 185
 - mappel::Gauss1DsMAP, 213
 - mappel::Gauss1DsMLE, 242
 - mappel::Gauss1DsModel, 269
 - mappel::Gauss2DMap, 298
 - mappel::Gauss2DMLE, 328
 - mappel::Gauss2DModel, 357
 - mappel::Gauss2DsMAP, 388
 - mappel::Gauss2DsMLE, 420
 - mappel::Gauss2DsModel, 451
 - mappel::Gauss2DsxyMAP, 479
 - mappel::Gauss2DsxyModel, 505
- set_lbound
 - mappel::Gauss1DMap, 131
 - mappel::Gauss1DMLE, 159
 - mappel::Gauss1DModel, 186
 - mappel::Gauss1DsMAP, 214
 - mappel::Gauss1DsMLE, 242
 - mappel::Gauss1DsModel, 269
 - mappel::Gauss2DMap, 298
 - mappel::Gauss2DMLE, 328
 - mappel::Gauss2DModel, 357
 - mappel::Gauss2DsMAP, 388
 - mappel::Gauss2DsMLE, 420
 - mappel::Gauss2DsModel, 451
 - mappel::Gauss2DsxyMAP, 480
 - mappel::Gauss2DsxyModel, 505
 - mappel::MAPEstimator, 555
 - mappel::MCMCAdaptor1Ds, 597
 - mappel::MCMCAdaptor1D, 579
 - mappel::MCMCAdaptor2Ds, 634
 - mappel::MLEstimator, 653
 - mappel::PointEmitterModel, 700
- set_max_sigma
 - mappel::Gauss1DsMAP, 214
 - mappel::Gauss1DsMLE, 242
 - mappel::Gauss1DsModel, 269
 - mappel::Gauss2DsMAP, 388
 - mappel::Gauss2DsMLE, 420
 - mappel::Gauss2DsModel, 452
 - mappel::Gauss2DsxyModel, 505
- set_max_sigma_ratio
 - mappel::Gauss2DsMAP, 388
 - mappel::Gauss2DsMLE, 420
 - mappel::Gauss2DsModel, 452
 - mappel::Gauss2DsxyModel, 505
- set_mcmc_num_phases
 - mappel::Gauss1DMap, 131
 - mappel::Gauss1DMLE, 159
 - mappel::Gauss1DModel, 186
 - mappel::Gauss1DsMAP, 214
 - mappel::Gauss1DsMLE, 242
 - mappel::Gauss1DsModel, 270
 - mappel::Gauss2DMap, 298
 - mappel::Gauss2DMLE, 328
 - mappel::Gauss2DModel, 357
 - mappel::Gauss2DsMAP, 388
 - mappel::Gauss2DsMLE, 420
 - mappel::Gauss2DsModel, 451
 - mappel::Gauss2DsxyMAP, 479
 - mappel::Gauss2DsxyModel, 505

- mappel::Gauss2DMLE, 328
- mappel::Gauss2DModel, 357
- mappel::Gauss2DsMAP, 389
- mappel::Gauss2DsMLE, 421
- mappel::Gauss2DsModel, 452
- mappel::Gauss2DsxyMAP, 480
- mappel::MCMCAdaptor1Ds, 598
- mappel::MCMCAdaptor1D, 580
- mappel::MCMCAdaptor2Ds, 634
- mappel::MCMCAdaptor2D, 616
- mappel::MCMCAdaptorBase, 642
- set_mcmc_sigma_scale
 - mappel::Gauss1DMAP, 131
 - mappel::Gauss1DMLE, 159
 - mappel::Gauss1DModel, 186
 - mappel::Gauss1DsMAP, 214
 - mappel::Gauss1DsMLE, 242
 - mappel::Gauss1DsModel, 270
 - mappel::Gauss2DMAP, 298
 - mappel::Gauss2DMLE, 328
 - mappel::Gauss2DModel, 357
 - mappel::Gauss2DsMAP, 389
 - mappel::Gauss2DsMLE, 421
 - mappel::Gauss2DsModel, 452
 - mappel::Gauss2DsxyMAP, 480
 - mappel::MCMCAdaptor1Ds, 598
 - mappel::MCMCAdaptor1D, 580
 - mappel::MCMCAdaptor2Ds, 634
 - mappel::MCMCAdaptor2D, 616
 - mappel::MCMCAdaptorBase, 642
- set_min_sigma
 - mappel::Gauss1DsMAP, 214, 215
 - mappel::Gauss1DsMLE, 243
 - mappel::Gauss1DsModel, 270
 - mappel::Gauss2DsMAP, 389
 - mappel::Gauss2DsMLE, 421
 - mappel::Gauss2DsModel, 452
 - mappel::Gauss2DsxyModel, 505
- set_model
 - mappel::CGaussHeuristicEstimator, 93
 - mappel::CGaussMLE, 102
 - mappel::Estimator, 110
 - mappel::HeuristicEstimator, 517
 - mappel::IterativeMaximizer, 541
 - mappel::NewtonDiagonalMaximizer, 668
 - mappel::NewtonMaximizer, 681
 - mappel::QuasiNewtonMaximizer, 733
 - mappel::SimulatedAnnealingMaximizer, 743
 - mappel::ThreadedEstimator, 775
 - mappel::TrustRegionMaximizer, 788
- set_param_names
 - mappel::Gauss1DMAP, 131
 - mappel::Gauss1DMLE, 159
 - mappel::Gauss1DModel, 186
- mappel::Gauss1DsMAP, 215
- mappel::Gauss1DsMLE, 243
- mappel::Gauss1DsModel, 270
- mappel::Gauss2DMAP, 298
- mappel::Gauss2DMLE, 328
- mappel::Gauss2DModel, 357
- mappel::Gauss2DsMAP, 389
- mappel::Gauss2DsMLE, 421
- mappel::Gauss2DsModel, 452
- mappel::Gauss2DsxyMAP, 480
- mappel::Gauss2DsxyModel, 505
- mappel::MAPEstimator, 556
- mappel::MCMCAdaptor1Ds, 598
- mappel::MCMCAdaptor1D, 580
- mappel::MCMCAdaptor2Ds, 634
- mappel::MCMCAdaptor2D, 616
- mappel::MLEstimator, 654
- mappel::PointEmitterModel, 700
- set_prior
 - mappel::Gauss1DMAP, 131
 - mappel::Gauss1DMLE, 159
 - mappel::Gauss1DModel, 186
 - mappel::Gauss1DsMAP, 215
 - mappel::Gauss1DsMLE, 243
 - mappel::Gauss1DsModel, 270
 - mappel::Gauss2DMAP, 299
 - mappel::Gauss2DMLE, 329
 - mappel::Gauss2DModel, 358
 - mappel::Gauss2DsMAP, 389
 - mappel::Gauss2DsMLE, 421
 - mappel::Gauss2DsModel, 453
 - mappel::Gauss2DsxyMAP, 480
 - mappel::Gauss2DsxyModel, 505
 - mappel::MAPEstimator, 556
 - mappel::MCMCAdaptor1Ds, 598
 - mappel::MCMCAdaptor1D, 580
 - mappel::MCMCAdaptor2Ds, 634, 635
 - mappel::MCMCAdaptor2D, 616
 - mappel::MLEstimator, 654
 - mappel::PointEmitterModel, 700
- set_psf_sigma
 - mappel::Gauss1DMAP, 132
 - mappel::Gauss1DMLE, 160
 - mappel::Gauss1DModel, 187
 - mappel::Gauss2DMAP, 299
 - mappel::Gauss2DMLE, 329
 - mappel::Gauss2DModel, 358
 - mappel::Gauss2DsxyMAP, 481
- set_rng_seed
 - mappel::Gauss1DMAP, 132
 - mappel::Gauss1DMLE, 160
 - mappel::Gauss1DModel, 187
 - mappel::Gauss1DsMAP, 215
 - mappel::Gauss1DsMLE, 243

- mappel::Gauss1DsModel, 271
- mappel::Gauss2DModel, 299
- mappel::Gauss2DModel, 329
- mappel::Gauss2DsMAP, 358
- mappel::Gauss2DsMAP, 390
- mappel::Gauss2DsMLE, 422
- mappel::Gauss2DsModel, 453
- mappel::Gauss2DsxyMAP, 481
- mappel::Gauss2DsxyModel, 505
- mappel::MAPEstimator, 556
- mappel::MCMCAdaptor1Ds, 598
- mappel::MCMCAdaptor1D, 580
- mappel::MCMCAdaptor2Ds, 635
- mappel::MCMCAdaptor2D, 616
- mappel::MLEstimator, 654
- mappel::PointEmitterModel, 700
- set_size
 - mappel::Gauss1DModel, 132
 - mappel::Gauss1DModel, 160
 - mappel::Gauss1DModel, 187
 - mappel::Gauss1DsMAP, 215
 - mappel::Gauss1DsMLE, 243, 244
 - mappel::Gauss1DsModel, 271
 - mappel::Gauss2DModel, 299
 - mappel::Gauss2DModel, 329
 - mappel::Gauss2DModel, 358
 - mappel::Gauss2DsMAP, 390
 - mappel::Gauss2DsMLE, 422
 - mappel::Gauss2DsModel, 453
 - mappel::Gauss2DsxyMAP, 481
 - mappel::Gauss2DsxyModel, 505
 - mappel::ImageFormat1DBase, 524
 - mappel::ImageFormat2DBase, 531
 - mappel::PoissonNoise1DObjective, 712
 - mappel::PoissonNoise2DObjective, 719
- set_stencil
 - mappel::IterativeMaximizer::MaximizerData, 564
- set_ubound
 - mappel::Gauss1DModel, 132
 - mappel::Gauss1DModel, 160
 - mappel::Gauss1DModel, 187
 - mappel::Gauss1DsMAP, 216
 - mappel::Gauss1DsMLE, 244
 - mappel::Gauss1DsModel, 271
 - mappel::Gauss2DModel, 299
 - mappel::Gauss2DModel, 329
 - mappel::Gauss2DModel, 358
 - mappel::Gauss2DsMAP, 390
 - mappel::Gauss2DsMLE, 422
 - mappel::Gauss2DsModel, 453
 - mappel::Gauss2DsxyMAP, 481
 - mappel::Gauss2DsxyModel, 506
 - mappel::MAPEstimator, 556
 - mappel::MCMCAdaptor1Ds, 598
 - mappel::MCMCAdaptor1D, 581
 - mappel::MCMCAdaptor2Ds, 635
 - mappel::MCMCAdaptor2D, 617
 - mappel::MLEstimator, 654
 - mappel::PointEmitterModel, 700
- setGlobalDefaultStrategy
 - omp_exception_catcher::impl_::OMPEXCEPTION←
Catcher, 688
- sgn
 - mappel, 46
- sigma
 - mappel::Gauss1DsModel::Stencil, 751
- sigma_ratio
 - mappel::Gauss2DsModel::Stencil, 760
- sigma_scale
 - mappel::Gauss1DModel, 138
 - mappel::Gauss1DModel, 166
 - mappel::Gauss1DModel, 193
 - mappel::Gauss1DsMAP, 221
 - mappel::Gauss1DsMLE, 249
 - mappel::Gauss1DsModel, 276
 - mappel::Gauss2DModel, 305
 - mappel::Gauss2DModel, 335
 - mappel::Gauss2DModel, 364
 - mappel::Gauss2DsMAP, 396
 - mappel::Gauss2DsMLE, 428
 - mappel::Gauss2DsModel, 459
 - mappel::Gauss2DsxyMAP, 487
 - mappel::MCMCAdaptor1Ds, 603
 - mappel::MCMCAdaptor1D, 585
 - mappel::MCMCAdaptor2Ds, 640
 - mappel::MCMCAdaptor2D, 621
 - mappel::MCMCAdaptorBase, 643
- sigmaX
 - mappel::Gauss2DsModel::Stencil, 760
 - mappel::Gauss2DsxyModel::Stencil, 755
- sigmaY
 - mappel::Gauss2DsModel::Stencil, 761
 - mappel::Gauss2DsxyModel::Stencil, 755
- simulate_image
 - mappel::methods, 59, 60
 - PoissonGaussianNoise2DObjective.h, 843, 844
- simulate_image_from_model
 - mappel::methods, 60, 61
- simulate_image_stack
 - mappel, 46
 - mappel::methods::openmp, 84
- SimulatedAnnealingMaximizer
 - mappel::SimulatedAnnealingMaximizer, 739
- size
 - mappel::Gauss1DModel, 138
 - mappel::Gauss1DModel, 166
 - mappel::Gauss1DModel, 193
 - mappel::Gauss1DsMAP, 221

- mappel::Gauss1DsMLE, [250](#)
- mappel::Gauss1DsModel, [276](#)
- mappel::Gauss2DMap, [306](#)
- mappel::Gauss2DMLE, [336](#)
- mappel::Gauss2DModel, [364](#)
- mappel::Gauss2DsMap, [396](#)
- mappel::Gauss2DsMLE, [428](#)
- mappel::Gauss2DsModel, [459](#)
- mappel::Gauss2DsxyMap, [487](#)
- mappel::Gauss2DsxyModel, [510](#)
- mappel::ImageFormat1DBase, [525](#)
- mappel::ImageFormat2DBase, [532](#)
- mappel::PoissonNoise1DObjective, [713](#)
- mappel::PoissonNoise2DObjective, [720](#)
- solve_TR_subproblem
 - mappel::TrustRegionMaximizer, [788](#)
- solve_restricted_step_length_newton
 - mappel::TrustRegionMaximizer, [788](#)
- square
 - mappel, [46](#)
- stackcomp.h, [851](#)
- StatsT
 - mappel, [27](#)
- Stencil
 - mappel::Gauss1DModel::Stencil, [747](#)
 - mappel::Gauss1DsModel::Stencil, [750](#)
 - mappel::Gauss2DModel::Stencil, [765](#)
 - mappel::Gauss2DsModel::Stencil, [759](#)
 - mappel::Gauss2DsxyModel::Stencil, [755](#)
- stencil
 - mappel::IterativeMaximizer::MaximizerData, [564](#)
- stencil.cpp, [852](#)
- stencil.h, [853](#)
- StencilVecT
 - mappel, [27](#)
 - mappel::Gauss1DMap, [117](#)
 - mappel::Gauss1DMLE, [145](#)
 - mappel::Gauss1DModel, [173](#)
 - mappel::Gauss1DsMap, [200](#)
 - mappel::Gauss1DsMLE, [228](#)
 - mappel::Gauss1DsModel, [256](#)
 - mappel::Gauss2DMap, [284](#)
 - mappel::Gauss2DMLE, [314](#)
 - mappel::Gauss2DModel, [343](#)
 - mappel::Gauss2DsMap, [372](#)
 - mappel::Gauss2DsMLE, [404](#)
 - mappel::Gauss2DsModel, [436](#)
 - mappel::Gauss2DsxyMap, [467](#)
 - mappel::Gauss2DsxyModel, [493](#)
- StencilT
 - mappel, [27](#)
- step
 - mappel::IterativeMaximizer::MaximizerData, [567](#)
- StepSize
 - mappel::CGaussHeuristicEstimator, [89](#)
 - mappel::CGaussMLE, [97](#)
 - mappel::Estimator, [105](#)
 - mappel::HeuristicEstimator, [513](#)
 - mappel::IterativeMaximizer, [535](#)
 - mappel::NewtonDiagonalMaximizer, [661](#)
 - mappel::NewtonMaximizer, [674](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::SimulatedAnnealingMaximizer, [739](#)
 - mappel::ThreadedEstimator, [771](#)
 - mappel::TrustRegionMaximizer, [780](#)
- Strategy
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [687](#)
- StringVecT
 - mappel, [27](#)
- Success
 - mappel::CGaussHeuristicEstimator, [89](#)
 - mappel::CGaussMLE, [97](#)
 - mappel::Estimator, [105](#)
 - mappel::HeuristicEstimator, [513](#)
 - mappel::IterativeMaximizer, [535](#)
 - mappel::NewtonDiagonalMaximizer, [661](#)
 - mappel::NewtonMaximizer, [674](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::SimulatedAnnealingMaximizer, [739](#)
 - mappel::ThreadedEstimator, [771](#)
 - mappel::TrustRegionMaximizer, [780](#)
- T_init
 - mappel::SimulatedAnnealingMaximizer, [745](#)
- TERM_BLACK
 - mappel, [47](#)
- TERM_BLUE
 - mappel, [47](#)
- TERM_CYAN
 - mappel, [47](#)
- TERM_DIM_BLACK
 - mappel, [47](#)
- TERM_DIM_BLUE
 - mappel, [47](#)
- TERM_DIM_CYAN
 - mappel, [47](#)
- TERM_DIM_GREEN
 - mappel, [47](#)
- TERM_DIM_MAGENTA
 - mappel, [48](#)
- TERM_DIM_RED
 - mappel, [48](#)
- TERM_DIM_WHITE
 - mappel, [48](#)
- TERM_DIM_YELLOW
 - mappel, [48](#)
- TERM_GREEN

- mappel, 48
- TERM_MAGENTA
 - mappel, 48
- TERM_RED
 - mappel, 48
- TERM_WHITE
 - mappel, 48
- TERM_YELLOW
 - mappel, 49
- theta
 - mappel::Gauss1DModel::Stencil, 749
 - mappel::Gauss1DsModel::Stencil, 753
 - mappel::Gauss2DModel::Stencil, 768
 - mappel::Gauss2DsModel::Stencil, 763
 - mappel::Gauss2DsxyModel::Stencil, 758
 - mappel::IterativeMaximizer::MaximizerData, 564
- theta_in_bounds
 - mappel::Gauss1DMap, 132
 - mappel::Gauss1DMLE, 160
 - mappel::Gauss1DModel, 187
 - mappel::Gauss1DsMap, 216
 - mappel::Gauss1DsMLE, 244
 - mappel::Gauss1DsModel, 271
 - mappel::Gauss2DMap, 299
 - mappel::Gauss2DMLE, 329
 - mappel::Gauss2DModel, 358
 - mappel::Gauss2DsMap, 390
 - mappel::Gauss2DsMLE, 422
 - mappel::Gauss2DsModel, 453
 - mappel::Gauss2DsxyMap, 481
 - mappel::Gauss2DsxyModel, 506
 - mappel::MAPEstimator, 556
 - mappel::MCMCAdaptor1Ds, 599
 - mappel::MCMCAdaptor1D, 581
 - mappel::MCMCAdaptor2Ds, 635
 - mappel::MCMCAdaptor2D, 617
 - mappel::MLEstimator, 654
 - mappel::PointEmitterModel, 701
- theta_seq
 - mappel::IterativeMaximizer::MaximizerData, 567
- theta_stack_in_bounds
 - mappel::Gauss1DMap, 133
 - mappel::Gauss1DMLE, 161
 - mappel::Gauss1DModel, 188
 - mappel::Gauss1DsMap, 216
 - mappel::Gauss1DsMLE, 244
 - mappel::Gauss1DsModel, 271
 - mappel::Gauss2DMap, 300
 - mappel::Gauss2DMLE, 330
 - mappel::Gauss2DModel, 359
 - mappel::Gauss2DsMap, 390
 - mappel::Gauss2DsMLE, 422
 - mappel::Gauss2DsModel, 453
 - mappel::Gauss2DsxyMap, 482
 - mappel::Gauss2DsxyModel, 506
 - mappel::MAPEstimator, 557
 - mappel::MCMCAdaptor1Ds, 599
 - mappel::MCMCAdaptor1D, 581
 - mappel::MCMCAdaptor2Ds, 635
 - mappel::MCMCAdaptor2D, 617
 - mappel::MLEstimator, 655
 - mappel::PointEmitterModel, 701
- thin_sample
 - mappel::mcmc, 50
- ThreadedEstimator
 - mappel::ThreadedEstimator, 771
- total_backtracks
 - mappel::IterativeMaximizer, 543
 - mappel::NewtonDiagonalMaximizer, 670
 - mappel::NewtonMaximizer, 683
 - mappel::QuasiNewtonMaximizer, 735
 - mappel::TrustRegionMaximizer, 792
- total_der_evals
 - mappel::IterativeMaximizer, 544
 - mappel::NewtonDiagonalMaximizer, 671
 - mappel::NewtonMaximizer, 684
 - mappel::QuasiNewtonMaximizer, 736
 - mappel::TrustRegionMaximizer, 792
- total_fun_evals
 - mappel::IterativeMaximizer, 544
 - mappel::NewtonDiagonalMaximizer, 671
 - mappel::NewtonMaximizer, 684
 - mappel::QuasiNewtonMaximizer, 736
 - mappel::TrustRegionMaximizer, 792
- total_iterations
 - mappel::IterativeMaximizer, 544
 - mappel::NewtonDiagonalMaximizer, 671
 - mappel::NewtonMaximizer, 684
 - mappel::QuasiNewtonMaximizer, 736
 - mappel::TrustRegionMaximizer, 792
- total_walltime
 - mappel::CGaussHeuristicEstimator, 95
 - mappel::CGaussMLE, 103
 - mappel::Estimator, 111
 - mappel::HeuristicEstimator, 519
 - mappel::IterativeMaximizer, 544
 - mappel::NewtonDiagonalMaximizer, 671
 - mappel::NewtonMaximizer, 684
 - mappel::QuasiNewtonMaximizer, 736
 - mappel::SimulatedAnnealingMaximizer, 745
 - mappel::ThreadedEstimator, 777
 - mappel::TrustRegionMaximizer, 792
- TrustRegionMaximizer
 - mappel::TrustRegionMaximizer, 781
- TrustRegionRadius
 - mappel::CGaussHeuristicEstimator, 89
 - mappel::CGaussMLE, 97
 - mappel::Estimator, 105

- mappel::HeuristicEstimator, [513](#)
 - mappel::IterativeMaximizer, [535](#)
 - mappel::NewtonDiagonalMaximizer, [661](#)
 - mappel::NewtonMaximizer, [674](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::SimulatedAnnealingMaximizer, [739](#)
 - mappel::ThreadedEstimator, [771](#)
 - mappel::TrustRegionMaximizer, [780](#)
- ubound
- mappel::Gauss1DModel, [138](#)
 - mappel::Gauss1DModel::Stencil, [138](#)
 - mappel::Gauss1DModel::MLE, [166](#)
 - mappel::Gauss1DsModel, [193](#)
 - mappel::Gauss1DsModel::Stencil, [222](#)
 - mappel::Gauss1DsModel::MLE, [250](#)
 - mappel::Gauss1DsModel::MLE, [250](#)
 - mappel::Gauss2DModel, [306](#)
 - mappel::Gauss2DModel::Stencil, [306](#)
 - mappel::Gauss2DsModel, [336](#)
 - mappel::Gauss2DsModel::Stencil, [365](#)
 - mappel::Gauss2DsModel::MLE, [397](#)
 - mappel::Gauss2DsModel::MLE, [429](#)
 - mappel::Gauss2DsModel::MLE, [429](#)
 - mappel::Gauss2DsModel::MLE, [460](#)
 - mappel::Gauss2DsModel::MLE, [487](#)
 - mappel::Gauss2DsModel::MLE, [511](#)
 - mappel::IterativeMaximizer::MaximizerData, [567](#)
 - mappel::MAPEstimator, [559](#)
 - mappel::MCMCAdaptor1Ds, [603](#)
 - mappel::MCMCAdaptor1Ds, [585](#)
 - mappel::MCMCAdaptor2Ds, [640](#)
 - mappel::MCMCAdaptor2Ds, [621](#)
 - mappel::MLEEstimator, [657](#)
 - mappel::PointEmitterModel, [703](#)
- Unassigned
- mappel::CGaussHeuristicEstimator, [89](#)
 - mappel::CGaussMLE, [97](#)
 - mappel::Estimator, [105](#)
 - mappel::HeuristicEstimator, [513](#)
 - mappel::IterativeMaximizer, [535](#)
 - mappel::NewtonDiagonalMaximizer, [661](#)
 - mappel::NewtonMaximizer, [674](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::SimulatedAnnealingMaximizer, [739](#)
 - mappel::ThreadedEstimator, [771](#)
 - mappel::TrustRegionMaximizer, [780](#)
- UniformDistT
- mappel, [27](#)
- update_internal_1D_estimators
- mappel::Gauss2DsModel, [506](#)
- update_internal_1Dsum_estimators
- mappel::Gauss2DModel, [300](#)
 - mappel::Gauss2DModel::MLE, [330](#)
 - mappel::Gauss2DModel::MLE, [359](#)
 - mappel::Gauss2DsModel, [390](#)
 - mappel::Gauss2DsModel::MLE, [422](#)
- mappel::Gauss2DsModel, [454](#)
- util.cpp, [855](#)
- util.h, [855](#)
- VecFieldT
- mappel, [27](#)
- VecT
- mappel, [27](#)
- X
- mappel::Gauss1DModel::Stencil, [749](#)
 - mappel::Gauss1DsModel::Stencil, [753](#)
 - mappel::Gauss2DModel::Stencil, [768](#)
 - mappel::Gauss2DsModel::Stencil, [763](#)
 - mappel::Gauss2DsModel::Stencil, [758](#)
- x
- mappel::Gauss1DModel::Stencil, [747](#)
 - mappel::Gauss1DsModel::Stencil, [751](#)
 - mappel::Gauss2DModel::Stencil, [766](#)
 - mappel::Gauss2DsModel::Stencil, [761](#)
 - mappel::Gauss2DsModel::Stencil, [755](#)
- x_model
- mappel::Gauss2DModel, [306](#)
 - mappel::Gauss2DModel::MLE, [336](#)
 - mappel::Gauss2DModel::MLE, [365](#)
 - mappel::Gauss2DsModel, [397](#)
 - mappel::Gauss2DsModel::MLE, [429](#)
 - mappel::Gauss2DsModel::MLE, [460](#)
 - mappel::Gauss2DsModel::MLE, [511](#)
- Y
- mappel::Gauss2DModel::Stencil, [768](#)
 - mappel::Gauss2DsModel::Stencil, [764](#)
 - mappel::Gauss2DsModel::Stencil, [758](#)
- y
- mappel::Gauss2DModel::Stencil, [766](#)
 - mappel::Gauss2DsModel::Stencil, [761](#)
 - mappel::Gauss2DsModel::Stencil, [756](#)
- y_model
- mappel::Gauss2DModel, [307](#)
 - mappel::Gauss2DModel::MLE, [337](#)
 - mappel::Gauss2DModel::MLE, [365](#)
 - mappel::Gauss2DsModel, [397](#)
 - mappel::Gauss2DsModel::MLE, [429](#)
 - mappel::Gauss2DsModel::MLE, [460](#)
 - mappel::Gauss2DsModel::MLE, [511](#)