

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	Constructor & Destructor Documentation	88
8.3.3	Member Function Documentation	88
8.3.4	Member Data Documentation	92
8.4	mappel::CGaussMLE< Model > Class Template Reference	93

8.4.1	Detailed Description	94
8.4.2	Constructor & Destructor Documentation	95
8.4.3	Member Function Documentation	95
8.4.4	Member Data Documentation	99
8.5	mappel::Estimator< Model > Class Template Reference	100
8.5.1	Detailed Description	101
8.5.2	Constructor & Destructor Documentation	101
8.5.3	Member Function Documentation	102
8.5.4	Friends And Related Function Documentation	105
8.5.5	Member Data Documentation	105
8.6	mappel::Gauss1DMP Class Reference	106
8.6.1	Detailed Description	110
8.6.2	Member Typedef Documentation	110
8.6.3	Constructor & Destructor Documentation	112
8.6.4	Member Function Documentation	113
8.6.5	Member Data Documentation	128
8.7	mappel::Gauss1DMLE Class Reference	134
8.7.1	Detailed Description	138
8.7.2	Member Typedef Documentation	138
8.7.3	Constructor & Destructor Documentation	140
8.7.4	Member Function Documentation	141
8.7.5	Member Data Documentation	156
8.8	mappel::Gauss1DModel Class Reference	162
8.8.1	Detailed Description	166
8.8.2	Member Typedef Documentation	166
8.8.3	Constructor & Destructor Documentation	168
8.8.4	Member Function Documentation	168
8.8.5	Member Data Documentation	183

8.9	mappel::Gauss1DsMAP Class Reference	189
8.9.1	Detailed Description	193
8.9.2	Member Typedef Documentation	193
8.9.3	Constructor & Destructor Documentation	195
8.9.4	Member Function Documentation	196
8.9.5	Member Data Documentation	211
8.10	mappel::Gauss1DsMLE Class Reference	217
8.10.1	Detailed Description	221
8.10.2	Member Typedef Documentation	222
8.10.3	Constructor & Destructor Documentation	224
8.10.4	Member Function Documentation	224
8.10.5	Member Data Documentation	239
8.11	mappel::Gauss1DsModel Class Reference	245
8.11.1	Detailed Description	249
8.11.2	Member Typedef Documentation	249
8.11.3	Constructor & Destructor Documentation	251
8.11.4	Member Function Documentation	252
8.11.5	Member Data Documentation	267
8.12	mappel::Gauss2DMAP Class Reference	272
8.12.1	Detailed Description	277
8.12.2	Member Typedef Documentation	277
8.12.3	Constructor & Destructor Documentation	279
8.12.4	Member Function Documentation	280
8.12.5	Member Data Documentation	295
8.13	mappel::Gauss2DMLE Class Reference	302
8.13.1	Detailed Description	307
8.13.2	Member Typedef Documentation	307
8.13.3	Constructor & Destructor Documentation	309

8.13.4	Member Function Documentation	310
8.13.5	Member Data Documentation	325
8.14	mappel::Gauss2DModel Class Reference	332
8.14.1	Detailed Description	336
8.14.2	Member Typedef Documentation	336
8.14.3	Constructor & Destructor Documentation	338
8.14.4	Member Function Documentation	339
8.14.5	Member Data Documentation	354
8.15	mappel::Gauss2DsMAP Class Reference	361
8.15.1	Detailed Description	365
8.15.2	Member Typedef Documentation	365
8.15.3	Constructor & Destructor Documentation	368
8.15.4	Member Function Documentation	368
8.15.5	Member Data Documentation	386
8.16	mappel::Gauss2DsMLE Class Reference	393
8.16.1	Detailed Description	397
8.16.2	Member Typedef Documentation	397
8.16.3	Constructor & Destructor Documentation	400
8.16.4	Member Function Documentation	400
8.16.5	Member Data Documentation	418
8.17	mappel::Gauss2DsModel Class Reference	425
8.17.1	Detailed Description	430
8.17.2	Member Typedef Documentation	430
8.17.3	Constructor & Destructor Documentation	432
8.17.4	Member Function Documentation	432
8.17.5	Member Data Documentation	449
8.18	mappel::Gauss2DsxyMAP Class Reference	456
8.18.1	Detailed Description	460

8.18.2	Member Typedef Documentation	460
8.18.3	Constructor & Destructor Documentation	462
8.18.4	Member Function Documentation	462
8.18.5	Member Data Documentation	477
8.19	mappel::Gauss2DsxyModel Class Reference	483
8.19.1	Detailed Description	487
8.19.2	Member Typedef Documentation	487
8.19.3	Constructor & Destructor Documentation	489
8.19.4	Member Function Documentation	489
8.19.5	Member Data Documentation	501
8.20	mappel::HeuristicEstimator< Model > Class Template Reference	506
8.20.1	Detailed Description	507
8.20.2	Constructor & Destructor Documentation	507
8.20.3	Member Function Documentation	507
8.20.4	Member Data Documentation	511
8.21	mappel::ImageFormat1DBase Class Reference	512
8.21.1	Detailed Description	513
8.21.2	Member Typedef Documentation	514
8.21.3	Constructor & Destructor Documentation	515
8.21.4	Member Function Documentation	515
8.21.5	Member Data Documentation	517
8.22	mappel::ImageFormat2DBase Class Reference	518
8.22.1	Detailed Description	520
8.22.2	Member Typedef Documentation	520
8.22.3	Constructor & Destructor Documentation	521
8.22.4	Member Function Documentation	522
8.22.5	Member Data Documentation	524
8.23	mappel::IterativeMaximizer< Model > Class Template Reference	525

8.23.1 Detailed Description	527
8.23.2 Member Enumeration Documentation	528
8.23.3 Constructor & Destructor Documentation	528
8.23.4 Member Function Documentation	528
8.23.5 Member Data Documentation	534
8.24 mappel::LogicalError Struct Reference	537
8.24.1 Detailed Description	537
8.24.2 Constructor & Destructor Documentation	538
8.25 mappel::MAPEstimator Class Reference	538
8.25.1 Detailed Description	540
8.25.2 Member Typedef Documentation	540
8.25.3 Constructor & Destructor Documentation	541
8.25.4 Member Function Documentation	541
8.25.5 Member Data Documentation	550
8.26 mappel::IterativeMaximizer< Model >::MaximizerData Class Reference	553
8.26.1 Detailed Description	554
8.26.2 Constructor & Destructor Documentation	554
8.26.3 Member Function Documentation	555
8.26.4 Member Data Documentation	558
8.27 mappel::MCMCAdaptor1D Class Reference	561
8.27.1 Detailed Description	563
8.27.2 Member Typedef Documentation	563
8.27.3 Constructor & Destructor Documentation	564
8.27.4 Member Function Documentation	564
8.27.5 Member Data Documentation	574
8.28 mappel::MCMCAdaptor1Ds Class Reference	578
8.28.1 Detailed Description	581
8.28.2 Member Typedef Documentation	581

8.28.3	Constructor & Destructor Documentation	581
8.28.4	Member Function Documentation	582
8.28.5	Member Data Documentation	591
8.29	mappel::MCMCAdaptor2D Class Reference	596
8.29.1	Detailed Description	599
8.29.2	Member Typedef Documentation	599
8.29.3	Constructor & Destructor Documentation	599
8.29.4	Member Function Documentation	600
8.29.5	Member Data Documentation	609
8.30	mappel::MCMCAdaptor2Ds Class Reference	614
8.30.1	Detailed Description	617
8.30.2	Member Typedef Documentation	617
8.30.3	Constructor & Destructor Documentation	617
8.30.4	Member Function Documentation	618
8.30.5	Member Data Documentation	628
8.31	mappel::MCMCAdaptorBase Class Reference	633
8.31.1	Detailed Description	633
8.31.2	Constructor & Destructor Documentation	633
8.31.3	Member Function Documentation	634
8.31.4	Member Data Documentation	635
8.32	mappel::MLEstimator Class Reference	635
8.32.1	Detailed Description	638
8.32.2	Member Typedef Documentation	638
8.32.3	Constructor & Destructor Documentation	638
8.32.4	Member Function Documentation	638
8.32.5	Member Data Documentation	647
8.33	mappel::ModelBoundsError Struct Reference	650
8.33.1	Detailed Description	650

8.33.2	Constructor & Destructor Documentation	650
8.34	mappel::NewtonDiagonalMaximizer< Model > Class Template Reference	651
8.34.1	Detailed Description	653
8.34.2	Member Typedef Documentation	653
8.34.3	Member Enumeration Documentation	653
8.34.4	Constructor & Destructor Documentation	653
8.34.5	Member Function Documentation	654
8.34.6	Member Data Documentation	660
8.35	mappel::NewtonMaximizer< Model > Class Template Reference	663
8.35.1	Detailed Description	665
8.35.2	Member Typedef Documentation	666
8.35.3	Member Enumeration Documentation	666
8.35.4	Constructor & Destructor Documentation	666
8.35.5	Member Function Documentation	666
8.35.6	Member Data Documentation	672
8.36	mappel::NotImplementedError Struct Reference	676
8.36.1	Detailed Description	676
8.36.2	Constructor & Destructor Documentation	676
8.37	mappel::NumericalError Struct Reference	676
8.37.1	Detailed Description	677
8.37.2	Constructor & Destructor Documentation	677
8.38	omp_exception_catcher::impl_::OMPExcptionCatcher< IntType > Class Template Reference	677
8.38.1	Detailed Description	678
8.38.2	Member Enumeration Documentation	678
8.38.3	Constructor & Destructor Documentation	678
8.38.4	Member Function Documentation	678
8.39	mappel::ParameterValueError Struct Reference	679
8.39.1	Detailed Description	680

8.39.2	Constructor & Destructor Documentation	680
8.40	mappel::PointEmitterModel Class Reference	680
8.40.1	Detailed Description	683
8.40.2	Member Typedef Documentation	683
8.40.3	Constructor & Destructor Documentation	683
8.40.4	Member Function Documentation	684
8.40.5	Member Data Documentation	692
8.41	PoissonGaussianNoise2DObjective< ModelBase > Class Template Reference	695
8.41.1	Detailed Description	696
8.41.2	Member Typedef Documentation	696
8.41.3	Constructor & Destructor Documentation	697
8.41.4	Member Data Documentation	697
8.42	mappel::PoissonNoise1DObjective Class Reference	697
8.42.1	Detailed Description	699
8.42.2	Member Typedef Documentation	699
8.42.3	Constructor & Destructor Documentation	701
8.42.4	Member Function Documentation	701
8.42.5	Member Data Documentation	703
8.43	mappel::PoissonNoise2DObjective Class Reference	705
8.43.1	Detailed Description	706
8.43.2	Member Typedef Documentation	706
8.43.3	Constructor & Destructor Documentation	708
8.43.4	Member Function Documentation	708
8.43.5	Member Data Documentation	711
8.44	mappel::PriorMAP1DObjective Class Reference	712
8.44.1	Detailed Description	713
8.44.2	Member Typedef Documentation	713
8.44.3	Member Function Documentation	713

8.44.4	Member Data Documentation	714
8.45	mappel::QuasiNewtonMaximizer< Model > Class Template Reference	715
8.45.1	Detailed Description	717
8.45.2	Member Typedef Documentation	717
8.45.3	Member Enumeration Documentation	717
8.45.4	Constructor & Destructor Documentation	718
8.45.5	Member Function Documentation	718
8.45.6	Member Data Documentation	724
8.46	mappel::SimulatedAnnealingMaximizer< Model > Class Template Reference	727
8.46.1	Detailed Description	729
8.46.2	Constructor & Destructor Documentation	729
8.46.3	Member Function Documentation	729
8.46.4	Member Data Documentation	733
8.47	mappel::Gauss1DModel::Stencil Class Reference	734
8.47.1	Detailed Description	735
8.47.2	Member Typedef Documentation	735
8.47.3	Constructor & Destructor Documentation	736
8.47.4	Member Function Documentation	736
8.47.5	Friends And Related Function Documentation	737
8.47.6	Member Data Documentation	737
8.48	mappel::Gauss1DsModel::Stencil Class Reference	738
8.48.1	Detailed Description	739
8.48.2	Member Typedef Documentation	739
8.48.3	Constructor & Destructor Documentation	739
8.48.4	Member Function Documentation	740
8.48.5	Friends And Related Function Documentation	741
8.48.6	Member Data Documentation	741
8.49	mappel::Gauss2DsxyModel::Stencil Class Reference	742

8.49.1 Detailed Description	743
8.49.2 Member Typedef Documentation	743
8.49.3 Constructor & Destructor Documentation	744
8.49.4 Member Function Documentation	744
8.49.5 Friends And Related Function Documentation	745
8.49.6 Member Data Documentation	745
8.50 mappel::Gauss2DsModel::Stencil Class Reference	747
8.50.1 Detailed Description	748
8.50.2 Member Typedef Documentation	748
8.50.3 Constructor & Destructor Documentation	748
8.50.4 Member Function Documentation	749
8.50.5 Friends And Related Function Documentation	750
8.50.6 Member Data Documentation	750
8.51 mappel::Gauss2DModel::Stencil Class Reference	753
8.51.1 Detailed Description	754
8.51.2 Member Typedef Documentation	754
8.51.3 Constructor & Destructor Documentation	754
8.51.4 Member Function Documentation	755
8.51.5 Friends And Related Function Documentation	756
8.51.6 Member Data Documentation	756
8.52 mappel::ThreadedEstimator< Model > Class Template Reference	758
8.52.1 Detailed Description	759
8.52.2 Constructor & Destructor Documentation	759
8.52.3 Member Function Documentation	760
8.52.4 Member Data Documentation	763
8.53 mappel::TrustRegionMaximizer< Model > Class Template Reference	765
8.53.1 Detailed Description	767
8.53.2 Member Typedef Documentation	767
8.53.3 Member Enumeration Documentation	768
8.53.4 Constructor & Destructor Documentation	768
8.53.5 Member Function Documentation	768
8.53.6 Member Data Documentation	776

9 File Documentation	780
9.1 display.cpp File Reference	780
9.1.1 Detailed Description	781
9.2 display.h File Reference	781
9.2.1 Detailed Description	782
9.3 estimator.h File Reference	782
9.3.1 Detailed Description	783
9.3.2 Typedef Documentation	783
9.4 estimator_impl.h File Reference	783
9.4.1 Detailed Description	783
9.5 estimator_statics.cpp File Reference	784
9.5.1 Detailed Description	784
9.6 Gauss1DMAP.cpp File Reference	784
9.6.1 Detailed Description	784
9.7 Gauss1DMAP.h File Reference	785
9.7.1 Detailed Description	785
9.8 Gauss1DMLE.cpp File Reference	785
9.8.1 Detailed Description	786
9.9 Gauss1DMLE.h File Reference	786
9.9.1 Detailed Description	786
9.10 Gauss1DModel.cpp File Reference	787
9.10.1 Detailed Description	787
9.11 Gauss1DModel.h File Reference	787
9.11.1 Detailed Description	788
9.12 Gauss1DsMAP.cpp File Reference	788
9.12.1 Detailed Description	788
9.13 Gauss1DsMAP.h File Reference	788
9.13.1 Detailed Description	789

9.14 Gauss1DsMLE.cpp File Reference	789
9.14.1 Detailed Description	789
9.15 Gauss1DsMLE.h File Reference	790
9.15.1 Detailed Description	790
9.16 Gauss1DsModel.cpp File Reference	790
9.16.1 Detailed Description	791
9.17 Gauss1DsModel.h File Reference	791
9.17.1 Detailed Description	791
9.18 Gauss2DMP.cpp File Reference	792
9.18.1 Detailed Description	792
9.19 Gauss2DMP.h File Reference	792
9.19.1 Detailed Description	793
9.20 Gauss2DMLE.cpp File Reference	793
9.20.1 Detailed Description	793
9.21 Gauss2DMLE.h File Reference	793
9.21.1 Detailed Description	794
9.22 Gauss2DModel.cpp File Reference	794
9.22.1 Detailed Description	794
9.23 Gauss2DModel.h File Reference	795
9.23.1 Detailed Description	795
9.24 Gauss2DsMAP.cpp File Reference	796
9.24.1 Detailed Description	796
9.25 Gauss2DsMAP.h File Reference	796
9.25.1 Detailed Description	797
9.26 Gauss2DsMLE.cpp File Reference	797
9.26.1 Detailed Description	797
9.27 Gauss2DsMLE.h File Reference	797
9.27.1 Detailed Description	798

9.28 Gauss2DsModel.cpp File Reference	798
9.28.1 Detailed Description	798
9.29 Gauss2DsModel.h File Reference	799
9.29.1 Detailed Description	799
9.30 Gauss2DsxyMAP.h File Reference	800
9.30.1 Detailed Description	800
9.31 Gauss2DsxyModel.h File Reference	800
9.31.1 Detailed Description	801
9.32 ImageFormat1DBase.cpp File Reference	801
9.32.1 Detailed Description	802
9.33 ImageFormat1DBase.h File Reference	802
9.33.1 Detailed Description	802
9.34 ImageFormat2DBase.cpp File Reference	803
9.34.1 Detailed Description	803
9.35 ImageFormat2DBase.h File Reference	803
9.35.1 Detailed Description	804
9.36 Install.md File Reference	804
9.37 MAPEstimator.h File Reference	804
9.37.1 Detailed Description	805
9.38 mcmc.cpp File Reference	805
9.39 mcmc.h File Reference	806
9.39.1 Detailed Description	807
9.40 MCMCAdaptor1D.cpp File Reference	807
9.40.1 Detailed Description	807
9.41 MCMCAdaptor1D.h File Reference	807
9.41.1 Detailed Description	808
9.42 MCMCAdaptor1Ds.cpp File Reference	808
9.42.1 Detailed Description	808

9.43	MCMCAdaptor1Ds.h File Reference	809
9.43.1	Detailed Description	809
9.44	MCMCAdaptor2D.cpp File Reference	809
9.44.1	Detailed Description	809
9.45	MCMCAdaptor2D.h File Reference	810
9.45.1	Detailed Description	810
9.46	MCMCAdaptor2Ds.cpp File Reference	810
9.46.1	Detailed Description	810
9.47	MCMCAdaptor2Ds.h File Reference	811
9.47.1	Detailed Description	811
9.48	MCMCAdaptorBase.cpp File Reference	811
9.48.1	Detailed Description	811
9.49	MCMCAdaptorBase.h File Reference	812
9.49.1	Detailed Description	812
9.50	MLEstimator.h File Reference	812
9.50.1	Detailed Description	813
9.51	model_methods.h File Reference	814
9.52	model_methods_impl.h File Reference	817
9.53	numerical.cpp File Reference	820
9.53.1	Detailed Description	821
9.54	numerical.h File Reference	821
9.54.1	Detailed Description	822
9.55	OMPExcptionCatcher.h File Reference	822
9.55.1	Detailed Description	823
9.56	openmp_methods.h File Reference	823
9.56.1	Detailed Description	826
9.57	PointEmitterModel.cpp File Reference	826
9.57.1	Detailed Description	826

9.58	PointEmitterModel.h File Reference	827
9.58.1	Detailed Description	827
9.59	PoissonGaussianNoise2DObjective.cpp File Reference	828
9.60	PoissonGaussianNoise2DObjective.h File Reference	828
9.60.1	Detailed Description	829
9.60.2	Function Documentation	829
9.61	PoissonNoise1DObjective.cpp File Reference	831
9.61.1	Detailed Description	831
9.62	PoissonNoise1DObjective.h File Reference	832
9.62.1	Detailed Description	833
9.63	PoissonNoise2DObjective.cpp File Reference	833
9.63.1	Detailed Description	834
9.64	PoissonNoise2DObjective.h File Reference	834
9.64.1	Detailed Description	835
9.65	PriorMAP1DObjective.h File Reference	836
9.65.1	Detailed Description	836
9.66	README.md File Reference	836
9.67	rng.cpp File Reference	836
9.68	rng.h File Reference	837
9.68.1	Detailed Description	837
9.69	stackcomp.h File Reference	838
9.69.1	Detailed Description	839
9.70	stencil.cpp File Reference	839
9.70.1	Detailed Description	840
9.71	stencil.h File Reference	840
9.71.1	Detailed Description	841
9.72	util.cpp File Reference	842
9.73	util.h File Reference	842

1 MAPPEL

Mappel is an object-oriented image processing library for [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 has object-oriented interfaces in 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 emitter (molecule/protein) at effective resolutions 10-50 times smaller than the fundamental diffraction limit for optical microscopes. Operationally, this is the process of going from the blurry, noisy, pixelated images to the estimate of true emitter position and the estimate of the uncertainty in true position. Figure 1 shows the point emitter localization process visually utilizing realistic physical scales for a typical super-resolution fluorescence microscope configuration, and showing typical effective fitting resolution.

Figure 1: Effective fitting resolution in typical applications

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 interger 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](https://github.com/markjolah/OlahScienceGentooOverlay).

```
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.

- `MAPPET_PYTHON_VERSIONS` - List of python X.Y versions seperated by ";" to build modules for (e.g., "3.4;3.5;3.6")
- `MAPPET_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 >	100
mappel::ThreadedEstimator< Model >	758
mappel::CGaussHeuristicEstimator< Model >	87
mappel::CGaussMLE< Model >	93
mappel::HeuristicEstimator< Model >	506
mappel::IterativeMaximizer< Model >	525
mappel::NewtonDiagonalMaximizer< Model >	651
mappel::NewtonMaximizer< Model >	663
mappel::QuasiNewtonMaximizer< Model >	715
mappel::TrustRegionMaximizer< Model >	765
mappel::SimulatedAnnealingMaximizer< Model >	727
mappel::ImageFormat1DBase	512
mappel::Gauss1DModel	162
mappel::Gauss1DMAP	106
mappel::Gauss1DMLE	134
mappel::Gauss2DsxyMAP	456

mappel::Gauss1DsModel	245
mappel::Gauss1DsMAP	189
mappel::Gauss1DsMLE	217
mappel::PoissonNoise1DObjective	697
mappel::Gauss1DMAP	106
mappel::Gauss1DMLE	134
mappel::Gauss1DsMAP	189
mappel::Gauss1DsMLE	217
mappel::Gauss2DsxyMAP	456
ImageFormat1DBase	
PoissonGaussianNoise2DObjective< ModelBase >	695
mappel::ImageFormat2DBase	518
mappel::Gauss2DModel	332
mappel::Gauss2DMAP	272
mappel::Gauss2DMLE	302
mappel::Gauss2DsModel	425
mappel::Gauss2DsMAP	361
mappel::Gauss2DsMLE	393
mappel::Gauss2DsxyModel	483
mappel::PoissonNoise2DObjective	705
mappel::Gauss2DMAP	272
mappel::Gauss2DMLE	302
mappel::Gauss2DsMAP	361
mappel::Gauss2DsMLE	393
MappelError	
mappel::ArrayShapeError	85
mappel::ArraySizeError	86
mappel::LogicalError	537
mappel::ModelBoundsError	650
mappel::NotImplementedError	676

mappel::NumericalError	676
mappel::ParameterValueError	679
mappel::IterativeMaximizer< Model >::MaximizerData	553
mappel::MCMCAdaptorBase	633
mappel::MCMCAdaptor1D	561
mappel::Gauss1DModel	162
mappel::MCMCAdaptor1Ds	578
mappel::Gauss1DsModel	245
mappel::MCMCAdaptor2D	596
mappel::Gauss2DModel	332
mappel::MCMCAdaptor2Ds	614
mappel::Gauss2DsModel	425
omp_exception_catcher::impl_::OMPExceptionCatcher< IntType >	677
mappel::PointEmitterModel	680
mappel::Gauss1DModel	162
mappel::Gauss1DsModel	245
mappel::Gauss2DModel	332
mappel::Gauss2DsModel	425
mappel::Gauss2DsxyModel	483
mappel::MAPEstimator	538
mappel::Gauss1DMAP	106
mappel::Gauss1DsMAP	189
mappel::Gauss2DMAP	272
mappel::Gauss2DsMAP	361
mappel::Gauss2DsxyMAP	456
mappel::MCMCAdaptor1D	561
mappel::MLEstimator	635
mappel::Gauss1DMLE	134
mappel::Gauss1DsMLE	217
mappel::Gauss2DMLE	302

mappel::Gauss2DsMLE	393
mappel::PriorMAP1DObjective	712
mappel::Gauss1DModel::Stencil	734
mappel::Gauss1DsModel::Stencil	738
mappel::Gauss2DsxyModel::Stencil	742
mappel::Gauss2DsModel::Stencil	747
mappel::Gauss2DModel::Stencil	753

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 >	93
mappel::Estimator< Model >	100
mappel::Gauss1DMAP	
A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	106
mappel::Gauss1DMLE	
A 1D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective	134
mappel::Gauss1DModel	
A base class for 1D Gaussian PSF with a fixed sigma (standard dev.)	162
mappel::Gauss1DsMAP	
A 1D Gaussian with variable PSF sigma under an Poisson read noise assumption and MAP Objective	189
mappel::Gauss1DsMLE	
A 1D Gaussian with variable PSF under an Poisson noise assumption and maximum-likelihood estimator	217
mappel::Gauss1DsModel	
Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels	245

<code>mappel::Gauss2DMap</code>	A 2D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	272
<code>mappel::Gauss2DMLE</code>	A 2D Gaussian with fixed PSF under an Poisson noise assumption and maximum-likelihood objective	302
<code>mappel::Gauss2DModel</code>	A base class for 2D Gaussian PSF with fixed but possibly asymmetric sigma	332
<code>mappel::Gauss2DsMAP</code>	A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum a-posteriori objective	361
<code>mappel::Gauss2DsMLE</code>	A 2D Gaussian with a variable scalar PSF sigma under a Poisson noise assumption using a maximum-likelihood objective	393
<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>]	425
<code>mappel::Gauss2DsxyMAP</code>	A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective	456
<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>]	483
<code>mappel::HeuristicEstimator< Model ></code>		506
<code>mappel::ImageFormat1DBase</code>	A virtual base class for 2D image localization objectives	512
<code>mappel::ImageFormat2DBase</code>	A virtual base class for 2D image localization objectives	518
<code>mappel::IterativeMaximizer< Model ></code>		525
<code>mappel::LogicalError</code>	Failure of code or algorithm logic	537
<code>mappel::MAPEstimator</code>	A Mixin class to configure a for MLE estimation (null prior)	538
<code>mappel::IterativeMaximizer< Model >::MaximizerData</code>		553
<code>mappel::MCMCAdaptor1D</code>		561
<code>mappel::MCMCAdaptor1Ds</code>		578
<code>mappel::MCMCAdaptor2D</code>		596
<code>mappel::MCMCAdaptor2Ds</code>		614

mappel::MCMCAdaptorBase	633
mappel::MLEstimator A Mixin class to configure a for MLE estimation (null prior)	635
mappel::ModelBoundsError Access outside the model bounds is attempted	650
mappel::NewtonDiagonalMaximizer< Model >	651
mappel::NewtonMaximizer< Model >	663
mappel::NotImplementedError Feature not yet implemented	676
mappel::NumericalError Expected numerical condition does not hold	676
omp_exception_catcher::impl_::OMPExcptionCatcher< IntType >	677
mappel::ParameterValueError Parameter value is not valid	679
mappel::PointEmitterModel A virtual Base type for point emitter localization models	680
PoissonGaussianNoise2DObjective< ModelBase > A Base type for point emitter localization models that use 2d images	695
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	697
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	705
mappel::PriorMAP1DObjective A Mixin class to configure a Gauss1DModel for MAP estimation (default 1D prior)	712
mappel::QuasiNewtonMaximizer< Model >	715
mappel::SimulatedAnnealingMaximizer< Model >	727
mappel::Gauss1DModel::Stencil Stencil for 1D fixed-sigma models	734
mappel::Gauss1DsModel::Stencil Stencil for 1D variable-sigma models	738
mappel::Gauss2DsxyModel::Stencil Stencil for 2D free-sigma (astigmatic) models	742
mappel::Gauss2DsModel::Stencil Stencil for 2D scalar-sigma models	747

mappel::Gauss2DModel::Stencil	
Stencil for 2D fixed-sigma models	753
mappel::ThreadedEstimator< Model >	758
mappel::TrustRegionMaximizer< Model >	765

6 File Index

6.1 File List

Here is a list of all files with brief descriptions:

display.cpp	780
display.h	781
estimator.h	
The class declaration and inline and templated functions for the Estimator class hierarchy	782
estimator_impl.h	783
estimator_statics.cpp	784
Gauss1DMAP.cpp	
The class definition and template Specializations for Gauss1DMAP	784
Gauss1DMAP.h	
The class declaration and inline and templated functions for Gauss1DMAP	785
Gauss1DMLE.cpp	
The class definition and template Specializations for Gauss1DMLE	785
Gauss1DMLE.h	
The class declaration and inline and templated functions for Gauss1DMLE	786
Gauss1DModel.cpp	
The class definition and template Specializations for Gauss1DModel	787
Gauss1DModel.h	
The class declaration and inline and templated functions for Gauss1DModel	787
Gauss1DsMAP.cpp	
The class definition and template Specializations for Gauss1DsMAP	788
Gauss1DsMAP.h	
The class declaration and inline and templated functions for Gauss1DsMAP	788
Gauss1DsMLE.cpp	
The class definition and template Specializations for Gauss1DsMLE	789
Gauss1DsMLE.h	
The class declaration and inline and templated functions for Gauss1DsMLE	790

Gauss1DsModel.cpp	790
The class definition and template Specializations for Gauss1DsModel	
Gauss1DsModel.h	791
The class declaration and inline and templated functions for Gauss1DsModel	
Gauss2DMAP.cpp	792
The class definition and template Specializations for Gauss2DMAP	
Gauss2DMAP.h	792
The class declaration and inline and templated functions for Gauss2DMAP	
Gauss2DMLE.cpp	793
The class definition and template Specializations for Gauss2DMLE	
Gauss2DMLE.h	793
The class declaration and inline and templated functions for Gauss2DMLE	
Gauss2DModel.cpp	794
The class definition and template Specializations for Gauss2DModel	
Gauss2DModel.h	795
The class declaration and inline and templated functions for Gauss2DModel	
Gauss2DsMAP.cpp	796
The class definition and template Specializations for Gauss2DsMAP	
Gauss2DsMAP.h	796
The class declaration and inline and templated functions for Gauss2DsMAP	
Gauss2DsMLE.cpp	797
The class definition and template Specializations for Gauss2DsMLE	
Gauss2DsMLE.h	797
The class declaration and inline and templated functions for Gauss2DsMLE	
Gauss2DsModel.cpp	798
The class definition and template Specializations for Gauss2DsModel	
Gauss2DsModel.h	799
The class declaration and inline and templated functions for Gauss2DsModel	
Gauss2DsxyMAP.h	800
The class declaration and inline and templated functions for Gauss2DsxyMAP	
Gauss2DsxyModel.h	800
The class declaration and inline and templated functions for Gauss2DsxyModel	
ImageFormat1DBase.cpp	801
The class definition and template Specializations for ImageFormat1DBase	
ImageFormat1DBase.h	802
The class declaration and inline and templated functions for ImageFormat1DBase	
ImageFormat2DBase.cpp	803
The class definition and template Specializations for ImageFormat2DBase	

ImageFormat2DBase.h	
The class declaration and inline and templated functions for ImageFormat2DBase	803
MAPEstimator.h	
Class declaration and inline and templated functions for MAPEstimator	804
mcmc.cpp	805
mcmc.h	
Templated MCMC methods for posterior estimation	806
MCMCAdaptor1D.cpp	
The class definition and template Specializations for MCMCAdaptor1D	807
MCMCAdaptor1D.h	
The class declaration and inline and templated functions for MCMCAdaptor1D	807
MCMCAdaptor1Ds.cpp	
The class definition and template Specializations for MCMCAdaptor1Ds	808
MCMCAdaptor1Ds.h	
The class declaration and inline and templated functions for MCMCAdaptor1Ds	809
MCMCAdaptor2D.cpp	
The class definition and template Specializations for MCMCAdaptor2D	809
MCMCAdaptor2D.h	
The class declaration and inline and templated functions for MCMCAdaptor2D	810
MCMCAdaptor2Ds.cpp	
The class definition and template Specializations for MCMCAdaptor2Ds	810
MCMCAdaptor2Ds.h	
The class declaration and inline and templated functions for MCMCAdaptor2Ds	811
MCMCAdaptorBase.cpp	
The class definition and template Specializations for MCMCAdaptorBase	811
MCMCAdaptorBase.h	
The class declaration and inline and templated functions for MCMCAdaptorBase	812
MLEstimator.h	
Class declaration and inline and templated functions for MLEstimator	812
model_methods.h	814
model_methods_impl.h	817
numerical.cpp	
Numerical matrix operations	820
numerical.h	
Numerical matrix operations	821
OMPExceptionCatcher.h	
A lightweight class for managing C++ exception handling strategies for openMP methods	822

openmp_methods.h	Namespaces for OpenMP parallelized versions of the mappel::model namespace functions (external methods)	823
PointEmitterModel.cpp	The class definition and template Specializations for PointEmitterModel	826
PointEmitterModel.h	The class declaration and inline and templated functions for PointEmitterModel	827
PoissonGaussianNoise2DObjective.cpp		828
PoissonGaussianNoise2DObjective.h	The class declaration and inline and templated functions for PoissonGaussianNoise2DObjective	828
PoissonNoise1DObjective.cpp	The class definition and template Specializations for PoissonNoise1DObjective	831
PoissonNoise1DObjective.h	The class declaration and inline and templated functions for PoissonNoise1DObjective	832
PoissonNoise2DObjective.cpp	The class definition and template Specializations for PoissonNoise2DObjective	833
PoissonNoise2DObjective.h	The class declaration and inline and templated functions for PoissonNoise2DObjective	834
PriorMAP1DObjective.h	The class declaration and inline and templated functions for PriorMAP1DObjective	836
rng.cpp		836
rng.h	Random number generation usign sfmt	837
stackcomp.h	Data-parallel versions of core computational functions using OpenMP	838
stencil.cpp	The stencils for pixel based computations	839
stencil.h	The stencils for pixel based computations	840
util.cpp		842
util.h		842

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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT<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, StencilT<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, StencilT<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, StencilT<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, StencilT<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, StencilT<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, StencilT<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::ThreadedEstimator< Model >::clear_stats().

7.1.3.12 `template<class Model > std::enable_if<std::is_base_of<Gauss2DsxyModel,Model>::value, StencilT<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, StencilT<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()`, and `cholesky_make_positive_definite()`.

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::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 symmetric matrix, fill in the upper triangle by copying the lower triangle. This operation modifies the matrix.

Definition at line 30 of file numerical.cpp.

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 symmetric matrix, fill in the lower triangle by copying the upper triangle. This operation modifies the matrix.

Definition at line 13 of file numerical.cpp.

Referenced by `cholesky_convert_full_matrix()`.

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(), and cholesky_make_positive_definite().

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::Gauss1D↵
Model::Stencil::X.

Referenced by make_unique().

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

Definition at line 175 of file estimator_impl.h.

References mappel::Estimator< Model>::get_stats().

7.1.3.90 `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.

7.1.3.91 `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::IterativeMaximizer< Model >::compute_estimate()`.

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::Estimator< Model >::estimate_max()`, `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 & obsI)`

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 & obsI, 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 & obsI)`

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 & obsI, 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 theta). Note: Expected information is an average quantity and is independent of the data. Enabled for [PoissonNoise2DObjective](#).

Parameters

<i>model</i>	PolImageCoordTEmitterModel
<i>s</i>	Stencil at desired theta

Returns

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

Definition at line 83 of file PoissonNoise2DObjective.h.

References `mappel::ImageFormat2DBase::size`.

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 & rlh, 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

<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 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 &*rlh*, [MatT](#) &*obsI*, [MatT](#) &*sequence*, [VecT](#) &*sequence_riIh*, [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_init, ParamT< Model > & theta_est, double & rllh, MatT & obsI, 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 & obsI, 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::ThreadedEstimator< Model>::clear_stats()`, `mappel::CGaussMLE< Model>::compute_estimate()`, `mappel::Estimator< Model>::estimate_max()`, `mappel::Estimator< Model>::estimate_max_debug()`, `mappel::methods::likelihood_objective()`, `mappel::TrustRegionMaximizer< Model>::maximize()`, `mappel::HeuristicEstimator< Model>::name()`, `mappel::CGaussHeuristicEstimator< Model>::name()`, `mappel::CGaussMLE< Model>::name()`, `mappel::PriorMAP1DObjective::prior_log_likelihood()`, `relative_log_likelihood()`, `rllh()`, `mappel::methods::objective::openmp::rllh_stack()`, `mappel::mcmc::sample_posterior()`, `mappel::mcmc::sample_posterior_debug()`, `mappel::SimulatedAnnealingMaximizer< Model>::SimulatedAnnealingMaximizer()`, 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()`, 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_::OMPExcceptionCatcher< 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_::OMPExcceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl_::OMPExcceptionCatcher< 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()`.

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

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::OMPEXceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< 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::OMPEXceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< 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](#)

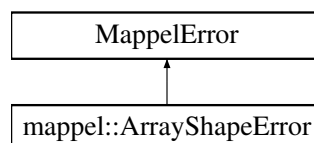
8 Class Documentation

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:

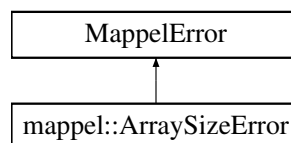
- [util.h](#)

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.

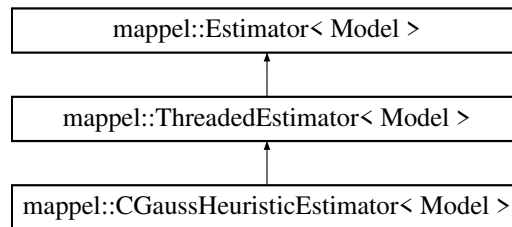
The documentation for this struct was generated from the following file:

- [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 Member Functions

- [CGaussHeuristicEstimator](#) (Model &model)
- std::string [name](#) () const
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im, const [ParamVecT](#)< Model > &theta_init, [ParamVecT](#)< Model > &theta, [VecT](#) &rllh, [CubeT](#) &obsl)
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im_stack, [ParamVecT](#)< Model > &theta_est, [VecT](#) &rllh_stack, [CubeT](#) &obsl_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)
- [StatsT](#) [get_stats](#) ()
- [StatsT](#) [get_debug_stats](#) ()
- 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)

Protected Member Functions

- 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.

8.3.1 Detailed Description

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

Definition at line 143 of file estimator.h.

8.3.2 Constructor & Destructor Documentation

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

Definition at line 145 of file estimator.h.

8.3.3 Member Function Documentation

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

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

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

Definition at line 287 of file estimator_impl.h.

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

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

```
8.3.3.2 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 142 of file estimator_impl.h.

References [mappel::methods::observed_information\(\)](#).

```
8.3.3.3 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 ) [inline], [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 192 of file estimator_impl.h.

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

```
8.3.3.4 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 151 of file estimator_impl.h.

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

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

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

Definition at line 59 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

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

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

Definition at line 77 of file estimator_impl.h.

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

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

Definition at line 68 of file estimator_impl.h.

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

8.3.3.8 `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 85 of file estimator_impl.h.

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

Definition at line 97 of file estimator_impl.h.

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

8.3.3.10 `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 106 of file estimator_impl.h.

8.3.3.11 `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 117 of file estimator_impl.h.

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

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

8.3.3.12 `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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

8.3.3.13 `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 222 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

8.3.3.14 `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 246 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::num_threads](#), [mappel::Estimator< Model >::record_walltime\(\)](#), [omp_exception_catcher::impl::OMPExcceptionCatcher< IntType >::rethrow\(\)](#), and [omp_exception_catcher::impl::OMPExcceptionCatcher< IntType >::run\(\)](#).

8.3.3.15 `template<class Model> StatsT mappel::ThreadedEstimator< Model >::get_debug_stats () [virtual], [inherited]`

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

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

Definition at line 281 of file estimator_impl.h.

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

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

Definition at line 45 of file estimator_impl.h.

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

8.3.3.17 `template<class Model> StatsT mappel::ThreadedEstimator< Model >::get_stats () [virtual], [inherited]`

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

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

Definition at line 271 of file estimator_impl.h.

References [mappel::Estimator< Model >::get_stats\(\)](#), [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::CGaussMLE< Model >::get_stats\(\)](#), and [mappel::IterativeMaximizer< Model >::get_stats\(\)](#).

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

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

Definition at line 147 of file estimator.h.

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

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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

8.3.4 Member Data Documentation

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

Definition at line 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model>::anneal()`, `mappel::IterativeMaximizer< Model>::backtrack()`, `mappel::ThreadedEstimator< Model>::clear_stats()`, `mappel::CGaussMLE< Model>::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate()`, `mappel::IterativeMaximizer< Model>::compute_estimate()`, `mappel::CGaussMLE< Model>::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, `mappel::IterativeMaximizer< Model>::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model>::maximize()`, `mappel::NewtonMaximizer< Model>::maximize()`, `mappel::QuasiNewtonMaximizer< Model>::maximize()`, and `mappel::TrustRegionMaximizer< Model>::maximize()`.

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 128 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()`.

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

Definition at line 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

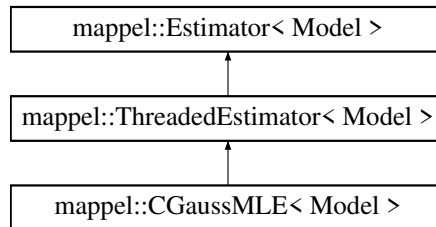
The documentation for this class was generated from the following files:

- [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 Member Functions

- `CGaussMLE` (`Model &model`, `int max_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`, `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 `max_iterations`

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)
 - 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.

8.4.1 Detailed Description

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

Definition at line 155 of file estimator.h.

8.4.2 Constructor & Destructor Documentation

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

Definition at line 158 of file estimator.h.

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

8.4.3 Member Function Documentation

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

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

Reimplemented in `mappel::IterativeMaximizer< Model >`.

Definition at line 287 of file estimator_impl.h.

References `mappel::cgauss_heuristic_compute_estimate()`, `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, and `mappel::methods::objective::rllh()`.

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

8.4.3.2 `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 142 of file estimator_impl.h.

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

8.4.3.3 `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 335 of file estimator_impl.h.

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

8.4.3.4 `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 344 of file estimator_impl.h.

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

8.4.3.5 `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 151 of file estimator_impl.h.

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

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

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

Definition at line 59 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

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

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

Definition at line 77 of file estimator_impl.h.

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

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

Definition at line 68 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

8.4.3.9 `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 85 of file estimator_impl.h.

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

Definition at line 97 of file estimator_impl.h.

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

8.4.3.11 `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 106 of file estimator_impl.h.

8.4.3.12 `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 117 of file estimator_impl.h.

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

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

8.4.3.13 `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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

8.4.3.14 `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 222 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

8.4.3.15 `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 246 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

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

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

Definition at line 326 of file `estimator_impl.h`.

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

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

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

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

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

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

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

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

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

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

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

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

References `mappel::Estimator< Model>::compute_estimate()`, `mappel::Estimator< Model>::compute_estimate_↵ debug()`, and `mappel::methods::objective::rllh()`.

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

Definition at line 203 of file `estimator_impl.h`.

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

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

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

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

8.4.4 Member Data Documentation

8.4.4.1 `template<class Model > int mappel::CGaussMLE< Model >::max_iterations`

Definition at line 157 of file estimator.h.

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

Definition at line 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::maximize()`.

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 128 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()`.

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

Definition at line 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

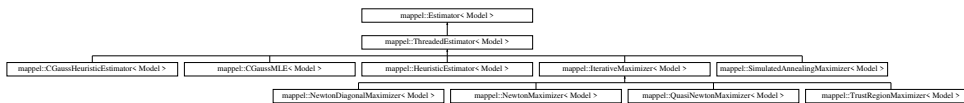
The documentation for this class was generated from the following files:

- [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 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](#) ()

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)

Protected Attributes

- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.

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 Constructor & Destructor Documentation

8.5.2.1 template<class Model > **mappel::Estimator< Model >::Estimator (Model &_model)** [inline]

Definition at line 39 of file estimator.h.

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

Definition at line 40 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\(\)](#).

8.5.3 Member Function Documentation

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

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

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

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

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

8.5.3.3 `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 142 of file `estimator_impl.h`.

References [mappel::methods::observed_information\(\)](#).

8.5.3.4 `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) [inline], [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 192 of file `estimator_impl.h`.

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

8.5.3.5 `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 151 of file estimator_impl.h.

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

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

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

Definition at line 59 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

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

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

Definition at line 77 of file estimator_impl.h.

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

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

Definition at line 68 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

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

Definition at line 85 of file estimator_impl.h.

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

Definition at line 97 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

8.5.3.11 `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 106 of file estimator_impl.h.

8.5.3.12 `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 117 of file estimator_impl.h.

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

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

8.5.3.13 `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>::~~Estimator()`.

8.5.3.14 `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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

8.5.3.15 `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()`.

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

Implemented in `mappel::IterativeMaximizer< Model>`, `mappel::CGaussMLE< Model>`, and `mappel::ThreadedEstimator< Model>`.

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

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

Definition at line 45 of file estimator_impl.h.

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

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

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

Definition at line 159 of file `estimator_impl.h`.

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

8.5.3.19 `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 >::compute_profile_estimate\(\)](#), and [mappel::Estimator< Model >::~~Estimator\(\)](#).

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

Definition at line 203 of file `estimator_impl.h`.

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

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

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

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

8.5.4 Friends And Related Function Documentation

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

8.5.5 Member Data Documentation

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

Definition at line 97 of file `estimator.h`.

Referenced by [mappel::SimulatedAnnealingMaximizer< Model >::anneal\(\)](#), [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::ThreadedEstimator< Model >::clear_stats\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_profile_estimate\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::IterativeMaximizer< Model >::local_maximize\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

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

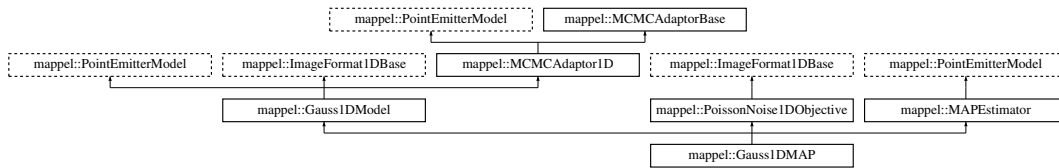
- [estimator.h](#)
- [estimator_impl.h](#)

8.6 mappel::Gauss1DMP Class Reference

A 1D Gaussian with fixed PSF under an Poisson Read Noise assumption and MAP Objective.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss1DMP.h>
```

Inheritance diagram for `mappel::Gauss1DMP`:



Public Types

- using `StencilVecT` = `std::vector< Stencil >`
- using `ParamT` = `arma::vec`
- using `ParamVecT` = `arma::mat`
- using `ImageCoordT` = `uint32_t`
- using `ImagePixelT` = `double`
- template<class CoordT >
using `ImageSizeShapeT` = `CoordT`
- template<class CoordT >
using `ImageSizeVecShapeT` = `arma::Col< CoordT >`
- using `ImageSizeT` = `ImageSizeShapeT< ImageCoordT >`
- using `ImageSizeVecT` = `ImageSizeVecShapeT< ImageCoordT >`
- template<class PixelT >
using `ImageShapeT` = `arma::Col< PixelT >`
- template<class PixelT >
using `ImageStackShapeT` = `arma::Mat< PixelT >`
- using `ImageT` = `ImageShapeT< ImagePixelT >`
- using `ImageStackT` = `ImageStackShapeT< ImagePixelT >`
- using `ModelDataT` = `ImageT`
- using `ModelDataStackT` = `ImageStackT`

Public Member Functions

- [Gauss1DMAP](#) (arma::Col< [ImageCoordT](#) > size, [VecT](#) psf_sigma, const std::string &prior_type=[DefaultPriorType](#))
- [Gauss1DMAP](#) ([ImageSizeT](#) size, double psf_sigma, const std::string &prior_type=[DefaultPriorType](#))
- [Gauss1DMAP](#) ([ImageSizeT](#) size, double psf_sigma, [CompositeDist](#) &&prior)
- [Gauss1DMAP](#) ([ImageSizeT](#) size, double psf_sigma, const [CompositeDist](#) &prior)
- [Gauss1DMAP](#) (const [Gauss1DMAP](#) &o)
- [Gauss1DMAP](#) & operator= (const [Gauss1DMAP](#) &o)
- [Gauss1DMAP](#) ([Gauss1DMAP](#) &&o)
- [Gauss1DMAP](#) & operator= ([Gauss1DMAP](#) &&o)
- double [get_psf_sigma](#) () const
- double [get_psf_sigma](#) ([IdxT](#) idx) const
- void [set_psf_sigma](#) (double new_psf_sigma)
- void [set_psf_sigma](#) (const [VecT](#) &new_psf_sigma)
- [StatsT](#) [get_stats](#) () const
- [Stencil](#) [make_stencil](#) (const [ParamT](#) &theta, bool compute_derivatives=true) const
Make a new Model::Stencil object at theta.
- double [pixel_model_value](#) ([IdxT](#) i, const [Stencil](#) &s) const
- void [pixel_grad](#) ([IdxT](#) i, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel_grad2](#) ([IdxT](#) i, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel_hess](#) ([IdxT](#) i, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel_hess_update](#) ([IdxT](#) i, const [Stencil](#) &s, double dm_ratio_m1, double dmm_ratio, [ParamT](#) &grad, [MatT](#) &hess) const
pixel derivative inner loop calculations.
- [Stencil](#) [initial_theta_estimate](#) (const [ImageT](#) &im) const
Fast, heuristic estimate of initial theta.
- [Stencil](#) [initial_theta_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta_init) const
- [IdxT](#) [get_num_params](#) () const
- void [check_param_shape](#) (const [ParamT](#) &theta) const
- void [check_param_shape](#) (const [ParamVecT](#) &theta) const
- void [check_psf_sigma](#) (double psf_sigma) const
- void [check_psf_sigma](#) (const [VecT](#) &psf_sigma) const
- [ParamT](#) [make_param](#) () const
- template<class FillT >
[ParamT](#) [make_param](#) (FillT fill) const
- [ParamVecT](#) [make_param_stack](#) ([IdxT](#) n) const
- template<class FillT >
[ParamVecT](#) [make_param_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT](#) [make_param_mat](#) () const
- template<class FillT >
[MatT](#) [make_param_mat](#) (FillT fill) const
- [CubeT](#) [make_param_mat_stack](#) ([IdxT](#) n) const
- template<class FillT >
[CubeT](#) [make_param_mat_stack](#) ([IdxT](#) n, FillT fill) const
- [CompositeDist](#) & [get_prior](#) ()
- const [CompositeDist](#) & [get_prior](#) () const
- void [set_prior](#) ([CompositeDist](#) &&prior_)
- void [set_prior](#) (const [CompositeDist](#) &prior_)
- [IdxT](#) [get_num_hyperparams](#) () const
- void [set_hyperparams](#) (const [VecT](#) &hyperparams)

- [VecT get_hyperparams](#) () const
- bool [has_hyperparam](#) (const std::string &name) const
- double [get_hyperparam_value](#) (const std::string &name) const
- int [get_hyperparam_index](#) (const std::string &name) const
- void [set_hyperparam_value](#) (const std::string &name, double value)
- void [rename_hyperparam](#) (const std::string &old_name, const std::string &new_name)
- [StringVecT get_param_names](#) () const
- void [set_param_names](#) (const [StringVecT](#) &desc)
- [StringVecT get_hyperparam_names](#) () const
- void [set_hyperparam_names](#) (const [StringVecT](#) &desc)
- template<class RngT >
 [ParamT sample_prior](#) (RngT &rng)
- [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](#)

8.6.1 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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by *mappel::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 306 of file PointEmitterModel.cpp.

References *mappel::PointEmitterModel::bounded_theta()*, and *mappel::PointEmitterModel::check_param_shape()*.

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

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

Definition at line 114 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::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().

8.6.4.34 `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.

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 97 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 91 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().

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

Definition at line 84 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 103 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 275 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 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, 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 212 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_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()`.

8.6.4.77 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 225 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 157 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 148 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 109 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 236 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 256 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 296 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 emitter intensity mean

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:

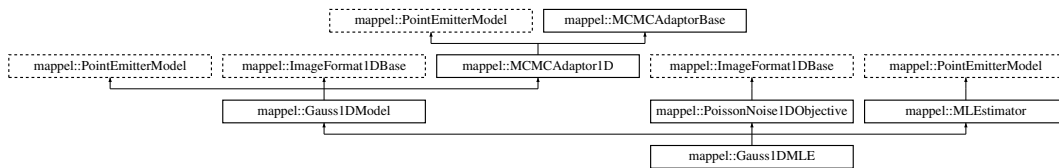
- [Gauss1DMap.h](#)
- [Gauss1DMap.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

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

Definition at line 114 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::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().

8.7.4.34 `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.

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_I, double kappa = default_intensity_kappa) [static],[inherited]`

Definition at line 97 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()`.

8.7.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos) [static],[inherited]`

Definition at line 91 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 84 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 103 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>	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()`.

8.7.4.57 `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 275 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 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, 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_l`, `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 212 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 225 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 157 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 148 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 109 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 236 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 256 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 296 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 emitter intensity mean

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:

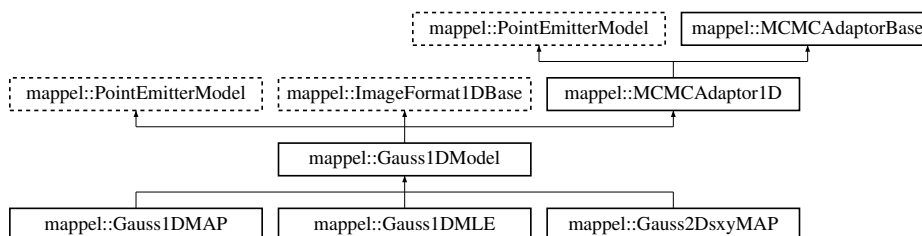
- [Gauss1DMLE.h](#)
- [Gauss1DMLE.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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.

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 97 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 91 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 84 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 103 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 275 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 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, 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_l`, `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 212 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_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()`.

8.8.4.77 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` [inherited]

Definition at line 225 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 157 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 148 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 109 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 236 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 256 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 296 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_l = INFINITY` `[static],[inherited]`

Default emitter intensity mean

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_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.8.5.7 `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()`, `make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

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:

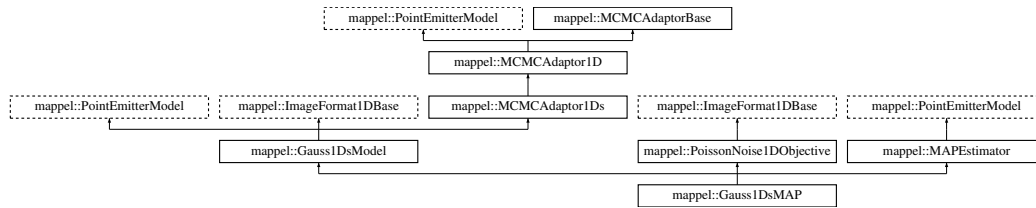
- [Gauss1DModel.h](#)
- [Gauss1DModel.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

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

Definition at line 114 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::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()`.

8.9.4.34 `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.

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_l, double kappa = default_intensity_kappa) [static],[inherited]`

Definition at line 97 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()`.

8.9.4.52 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos) [static],[inherited]`

Definition at line 91 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 84 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 103 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 275 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 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, 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 212 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_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().

8.9.4.77 void mappel::PointEmitterModel::set_lbound (const ParamT & *lbound*) [inherited]

Definition at line 225 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 157 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 148 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 109 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 236 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 256 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 296 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_l = INFINITY` `[static],[inherited]`

Default emitter intensity mean

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_l = 300` `[static],[inherited]`

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.9.5.7 `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_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()`.

8.9.5.14 `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:

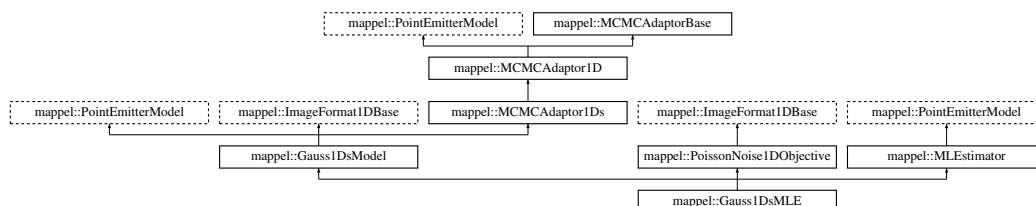
- [Gauss1DsMAP.h](#)
- [Gauss1DsMAP.cpp](#)

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_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](#)

8.10.1 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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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.

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 97 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::default_max_l`.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, `mappel::Gauss2DsModel::make_prior_beta_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `mappel::Gauss1DsModel::make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

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 91 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 84 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()`.

8.10.4.54 `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 103 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

<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()`.

8.10.4.57 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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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 212 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 225 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 157 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 148 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 109 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 236 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 256 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::make_stencil(), mappel::Gauss1DsModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DsxyModel::make_stencil(), mappel::Gauss2DsModel::make_stencil(), and mappel::PointEmitterModel::theta_stack_in_bounds().

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

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

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_l = INFINITY` [static],[inherited]

Default emitter intensity mean

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_l = 300` [static],[inherited]

Default emitter intensity mean

Definition at line 58 of file PointEmitterModel.h.

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

8.10.5.7 `const double mappel::PointEmitterModel::default_pixel_mean_bg = 4` [static],[inherited]

Default per-pixel mean background counts

Definition at line 61 of file PointEmitterModel.h.

Referenced by `mappel::Gauss1DsModel::make_default_prior_beta_position()`, `mappel::Gauss2DModel::make_default_prior_beta_position()`, `mappel::Gauss1DModel::make_default_prior_beta_position()`, `mappel::Gauss2DsModel::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, and `mappel::MCMCAdaptor1D::set_background_mcmc_sampling()`.

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:

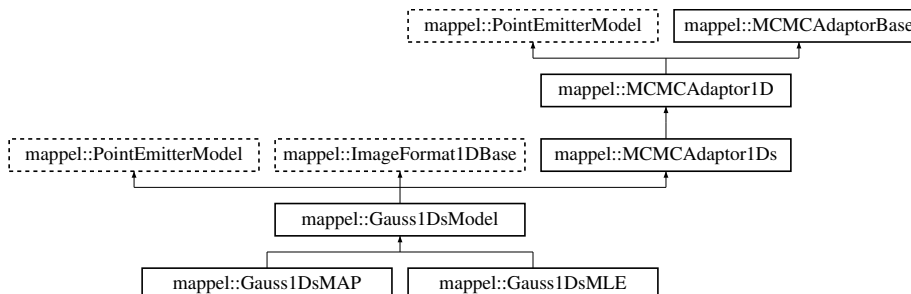
- [Gauss1DsMLE.h](#)
- [Gauss1DsMLE.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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.

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_I, double kappa_I, 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_I, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 97 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::default_max_I.

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 91 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 84 of file PointEmitterModel.cpp.

Referenced by `make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_default_prior_normal_position()`, `mappel::Gauss1DModel::make_default_prior_normal_position()`, `mappel::Gauss2DsModel::make_default_prior_normal_position()`, `mappel::Gauss2DModel::make_prior_normal_position()`, `make_prior_normal_position()`, `mappel::Gauss1DModel::make_prior_normal_position()`, and `mappel::Gauss2DsModel::make_prior_normal_position()`.

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 103 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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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 212 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 225 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 157 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 148 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 109 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 236 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 256 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 296 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 emitter intensity mean

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:

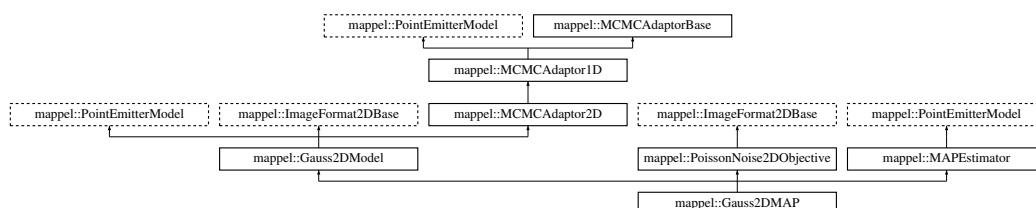
- [Gauss1DsModel.h](#)
- [Gauss1DsModel.cpp](#)

8.12 `mappel::Gauss2DMap` Class Reference

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

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DMap.h>
```

Inheritance diagram for `mappel::Gauss2DMap`:



Public Types

- using [Gauss1DSumModelT](#) = [Gauss1DMap](#)
- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32_t
- using [ImagePixelT](#) = double
- template<class CoordT >
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >
- using [ModelDataT](#) = [ImageT](#)
- using [ModelDataStackT](#) = [ImageStackT](#)

Public Member Functions

- [Gauss2DMap](#) ([ImageCoordT](#) size, double [psf_sigma](#), const std::string &prior_type=[DefaultPriorType](#))
- [Gauss2DMap](#) (const [ImageSizeT](#) &size, double [psf_sigma](#), const std::string &prior_type=[DefaultPriorType](#))
- template<class IntType , class FloatType >
[Gauss2DMap](#) (const arma::Col< IntType > &size, const arma::Col< FloatType > &[psf_sigma](#), const std::string &prior_type=[DefaultPriorType](#))
- [Gauss2DMap](#) (const [ImageSizeT](#) &size, const [VecT](#) &[psf_sigma](#), CompositeDist &&prior)
- [Gauss2DMap](#) ([ImageSizeT](#) &&size, [VecT](#) &&[psf_sigma](#), CompositeDist &&prior)
- [Gauss2DMap](#) (const [ImageSizeT](#) &size, const [VecT](#) &[psf_sigma](#), const CompositeDist &prior)
- [Gauss2DMap](#) (const [Gauss2DMap](#) &o)
- [Gauss2DMap](#) & operator= (const [Gauss2DMap](#) &o)
- [Gauss2DMap](#) ([Gauss2DMap](#) &&o)
- [Gauss2DMap](#) & operator= ([Gauss2DMap](#) &&o)
- void [set_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set_prior](#) (CompositeDist &&prior_)
- void [set_prior](#) (const CompositeDist &prior_)
- void [set_size](#) (const [ImageSizeT](#) &size_)
- const [VecT](#) & [get_psf_sigma](#) () const
- double [get_psf_sigma](#) (IdxT idx) const
- void [set_psf_sigma](#) (double new_psf_sigma)
- void [set_psf_sigma](#) (const [VecT](#) &new_psf_sigma)
- [StatsT](#) [get_stats](#) () const
- [Stencil](#) [make_stencil](#) (const [ParamT](#) &theta, bool compute_derivatives=true) const
Make a new Model::Stencil object at theta.
- double [pixel_model_value](#) (int i, int j, const [Stencil](#) &s) const
- void [pixel_grad](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad) const

- void `pixel_grad2` (int i, int j, const `Stencil` &s, `ParamT` &pgrad2) const
- void `pixel_hess` (int i, int j, const `Stencil` &s, `MatT` &hess) const
- void `pixel_hess_update` (int i, int j, const `Stencil` &s, double dm_ratio_m1, double dmm_ratio, `ParamT` &grad, `MatT` &hess) const
- pixel derivative inner loop calculations.*
- `Stencil initial_theta_estimate` (const `ImageT` &im)
- 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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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.

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 97 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::Gauss2Ds↵
 Model::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 91 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 84 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 103 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>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

Returns

A new [Stencil](#) object ready to compute with

Definition at line 131 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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`, `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 212 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 225 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 109 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 236 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 256 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()`.

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

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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 emitter intensity mean

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:

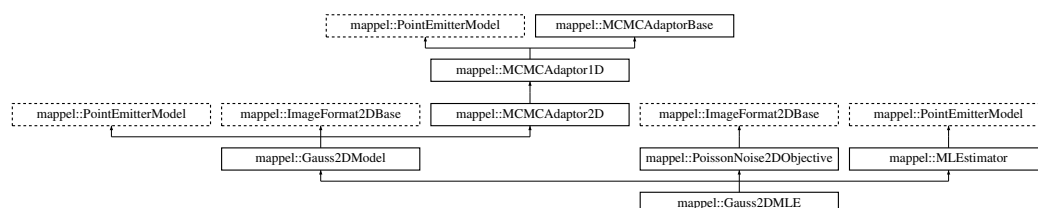
- [Gauss2DMAP.h](#)
- [Gauss2DMAP.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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.

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 97 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::Gauss2Ds↵
 Model::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 91 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 84 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 103 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>	Parameter to evaluate at
<i>compute_derivatives</i>	True to also prepare for derivative computations

Returns

A new [Stencil](#) object ready to compute with

Definition at line 131 of file Gauss2DModel.h.

References `mappel::Gauss2DModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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

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::l()`, `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::l()`, `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::l()`, `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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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`, `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 212 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 225 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 109 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 236 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 256 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 296 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 emitter intensity mean

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:

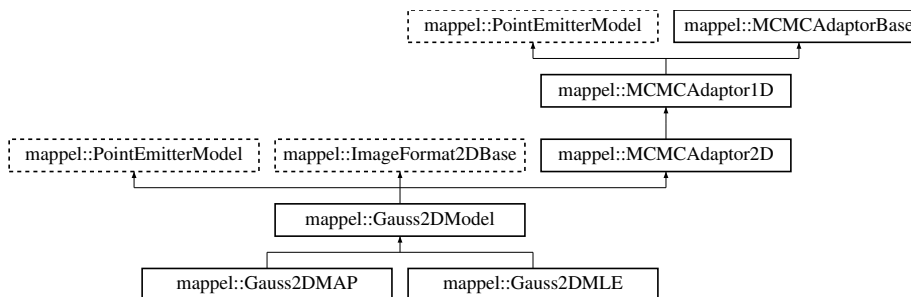
- [Gauss2DMLE.h](#)
- [Gauss2DMLE.cpp](#)

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 kappa_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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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.

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 97 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 91 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 84 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 103 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 275 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 316 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), 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`, `x`, `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 212 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 225 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 109 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 236 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 256 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 296 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 emitter intensity mean

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()`, `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()`, `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_` \leftrightarrow `sigma()`, `set_size()`, and `update_internal_1Dsum_estimators()`.

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

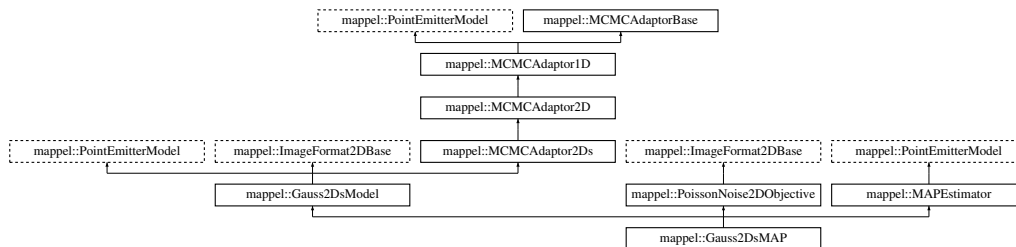
- [Gauss2DModel.h](#)
- [Gauss2DModel.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file `PointEmitterModel.cpp`.

References `mappel::rng_manager`.

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

Definition at line 114 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::Gauss2Ds←Model(), 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.

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 97 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 91 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 84 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 103 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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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::PointEmitterModel::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 212 of file PointEmitterModel.cpp.

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

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_l`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::PointEmitter←Model::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.15.4.85 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` [inherited]

Definition at line 225 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::Point←EmitterModel::get_ubound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::PointEmitterModel::set_ubound()`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

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

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 109 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::ImageFormat2DBase::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 236 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().

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

Definition at line 256 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::make_stencil(), mappel::Gauss1DsModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DsxyModel::make_stencil(), mappel::Gauss2DsModel::make_stencil(), and mappel::PointEmitterModel::theta_stack_in_bounds().

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

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

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::Gauss2DModel::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 emitter intensity mean

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::Gauss2←DsModel(), mappel::Gauss2DsModel::get_min_sigma(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel←::Gauss2DsModel::make_internal_1Dsum_estimator(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2Ds←Model::set_min_sigma(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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::PointEmitter←Model::get_stats(), mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::make_param(), mappel::Point←EmitterModel::make_param_mat(), mappel::PointEmitterModel::make_param_mat_stack(), mappel::PointEmitter←Model::make_param_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set←_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

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::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()`.

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

Definition at line 148 of file PointEmitterModel.h.

Referenced by `mappel::PointEmitterModel::bound_theta()`, `mappel::PointEmitterModel::bounded_theta()`, `mappel::PointEmitterModel::get_stats()`, `mappel::PointEmitterModel::get_ubound()`, `mappel::Gauss1DsModel::initial_theta_estimate()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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:

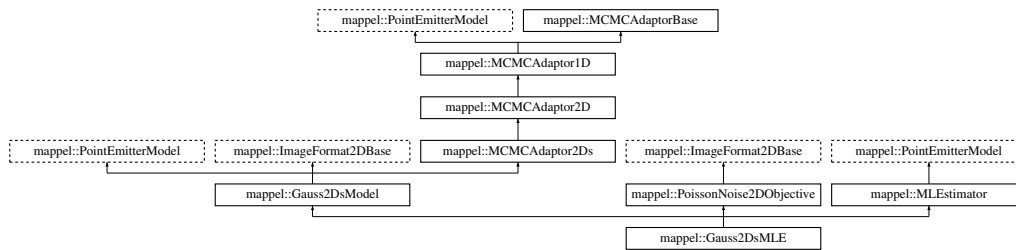
- [Gauss2DsMAP.h](#)
- [Gauss2DsMAP.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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::Gauss2Ds←Model(), 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.

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 97 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 91 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 84 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 103 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::l(), 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::l(), 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::l(), 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 275 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 316 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), 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::PointEmitterModel::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 212 of file PointEmitterModel.cpp.

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

Definition at line 65 of file MCMCAdaptor1D.cpp.

References `mappel::PointEmitterModel::default_mean_l`, `mappel::MCMCAdaptor1D::eta_l`, `mappel::PointEmitter←Model::get_hyperparam_value()`, and `mappel::MCMCAdaptorBase::sigma_scale`.

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

8.16.4.85 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` [inherited]

Definition at line 225 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::Point←EmitterModel::get_ubound()`, `mappel::Gauss1DsModel::set_max_sigma()`, `mappel::PointEmitterModel::set_ubound()`, `mappel::Gauss2DsModel::x_model`, and `mappel::Gauss2DsModel::y_model`.

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

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 109 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::ImageFormat2DBase::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 236 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()`.

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

Definition at line 256 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()`.

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

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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::Gauss2DModel::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 emitter intensity mean

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 θ_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 θ_{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 θ_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 θ_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::Gauss2←DsModel(), mappel::Gauss2DsModel::get_min_sigma(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel←::Gauss2DsModel::make_internal_1Dsum_estimator(), mappel::Gauss2DsModel::operator=(), mappel::Gauss2Ds←Model::set_min_sigma(), mappel::Gauss2DsModel::Stencil::sigmaX(), mappel::Gauss2DsModel::Stencil::sigmaY(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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::PointEmitter←Model::get_stats(), mappel::Gauss1DModel::initial_theta_estimate(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::PointEmitterModel::make_param(), mappel::Point←EmitterModel::make_param_mat(), mappel::PointEmitterModel::make_param_mat_stack(), mappel::PointEmitter←Model::make_param_stack(), mappel::PointEmitterModel::operator=(), mappel::PointEmitterModel::reflected_theta(), mappel::PointEmitterModel::set_bounds(), mappel::PointEmitterModel::set_lbound(), mappel::PointEmitterModel::set←_prior(), mappel::PointEmitterModel::set_ubound(), and mappel::PointEmitterModel::theta_in_bounds().

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:

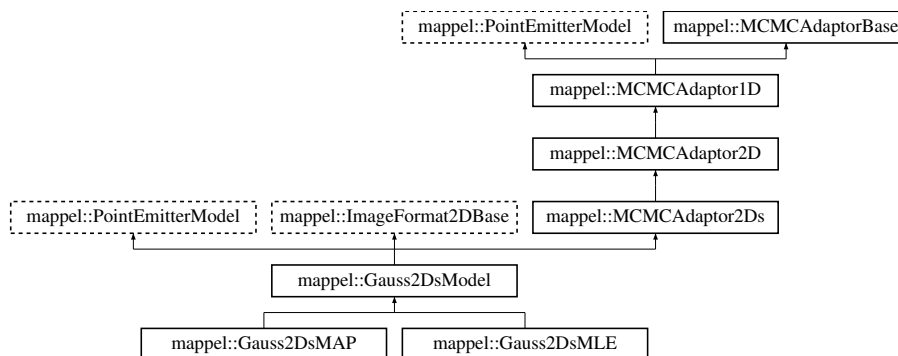
- [Gauss2DsMLE.h](#)
- [Gauss2DsMLE.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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.

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::Image↵
Format2DBase::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 97 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 91 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 84 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 103 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::l\(\)](#), [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::I()`, `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::I()`, `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::I()`, `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::I()`, `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 275 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 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, 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 212 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 225 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 109 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 236 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 256 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 296 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 emitter intensity mean

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:

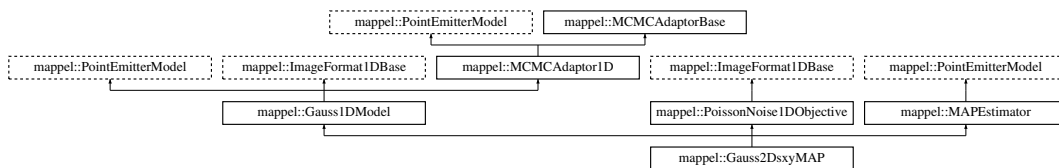
- [Gauss2DsModel.h](#)
- [Gauss2DsModel.cpp](#)

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_↵, 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 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::Gauss2DskyMAP::Gauss2DskyMAP (const ImageSizeT & size, const VecT & min_sigma, const VecT & max_sigma)

8.18.3.2 mappel::Gauss2DskyMAP::Gauss2DskyMAP (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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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`.

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 97 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 91 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 84 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 103 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 275 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 316 of file `PointEmitterModel.cpp`.

References `mappel::PointEmitterModel::check_param_shape()`, 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_l`, `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 212 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_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().

8.18.4.75 void mappel::PointEmitterModel::set_lbound (const ParamT & *lbound*) [inherited]

Definition at line 225 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 157 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 148 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 109 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 236 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 256 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()`.

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

Definition at line 296 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

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 emitter intensity mean

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:

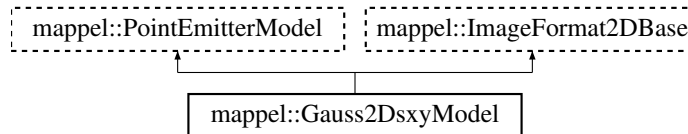
- [Gauss2DsxxyMAP.h](#)

8.19 mappel::Gauss2DsxxyModel Class Reference

A base class for 2D Gaussian PSF with axis-aligned gaussian with free parameters for both sigma_x and sigma_y. Gaussian sigma parameters sigma_x and sigma_y are measured in units of pixels. The model has 6 parameters, [x,y,l,bg,sigma_x,sigma_y].

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/Gauss2DsxxyModel.h>
```

Inheritance diagram for mappel::Gauss2DsxxyModel:



Classes

- class [Stencil](#)
Stencil for 2D free-sigma (astigmatic) models.

Public Types

- using [StencilVecT](#) = std::vector< [Stencil](#) >
- using [ParamT](#) = arma::vec
- using [ParamVecT](#) = arma::mat
- using [ImageCoordT](#) = uint32_t
- using [ImagePixelT](#) = double
- template<class CoordT >
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >

Public Member Functions

- [Gauss2DsxxyModel](#) (const [ImageSizeT](#) &size, const [VecT](#) &min_sigma, const [VecT](#) &max_sigma)
- void [set_hyperparams](#) (const [VecT](#) &hyperparams)
- void [set_prior](#) (CompositeDist &&prior_)
- void [set_size](#) (const [ImageSizeT](#) &size_)
- [VecT](#) [get_min_sigma](#) () const
- double [get_min_sigma](#) ([IdxT](#) dim) const
- [VecT](#) [get_max_sigma](#) () const
- double [get_max_sigma](#) ([IdxT](#) dim) const
- double [get_max_sigma_ratio](#) () const
- void [set_min_sigma](#) (const [VecT](#) &min_sigma)
- void [set_max_sigma](#) (const [VecT](#) &max_sigma)
- void [set_max_sigma_ratio](#) (double max_sigma_ratio)
- [StatsT](#) [get_stats](#) () const
- [Stencil](#) [make_stencil](#) (const [ParamT](#) &theta, bool compute_derivatives=true) const
Make a new Model::Stencil object at theta.
- double [pixel_model_value](#) (int i, int j, const [Stencil](#) &s) const
- void [pixel_grad](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad) const
- void [pixel_grad2](#) (int i, int j, const [Stencil](#) &s, [ParamT](#) &pgrad2) const
- void [pixel_hess](#) (int i, int j, const [Stencil](#) &s, [MatT](#) &hess) const
- void [pixel_hess_update](#) (int i, int j, const [Stencil](#) &s, double dm_ratio_m1, double dmm_ratio, [ParamT](#) &grad, [MatT](#) &hess) const
- [Stencil](#) [initial_theta_estimate](#) (const [ImageT](#) &im)
Fast, heuristic estimate of initial theta.
- [Stencil](#) [initial_theta_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta_init)
- [Stencil](#) [initial_theta_estimate](#) (const [ImageT](#) &im, const [ParamT](#) &theta_init, const std::string &estimator)
- void [sample_mcmc_candidate](#) (int sample_index, [ParamT](#) &candidate_theta, double scale=1.0)
- [IdxT](#) [get_num_params](#) () const
- void [check_param_shape](#) (const [ParamT](#) &theta) const
- void [check_param_shape](#) (const [ParamVecT](#) &theta) const
- void [check_psf_sigma](#) (double psf_sigma) const
- void [check_psf_sigma](#) (const [VecT](#) &psf_sigma) const
- [ParamT](#) [make_param](#) () const
- template<class FillT >
[ParamT](#) [make_param](#) (FillT fill) const
- [ParamVecT](#) [make_param_stack](#) ([IdxT](#) n) const
- template<class FillT >
[ParamVecT](#) [make_param_stack](#) ([IdxT](#) n, FillT fill) const
- [MatT](#) [make_param_mat](#) () const
- template<class FillT >
[MatT](#) [make_param_mat](#) (FillT fill) const
- [CubeT](#) [make_param_mat_stack](#) ([IdxT](#) n) const
- template<class FillT >
[CubeT](#) [make_param_mat_stack](#) ([IdxT](#) n, FillT fill) const
- CompositeDist & [get_prior](#) ()
- const CompositeDist & [get_prior](#) () const
- void [set_prior](#) (const CompositeDist &prior_)
- [IdxT](#) [get_num_hyperparams](#) () const
- [VecT](#) [get_hyperparams](#) () const
- bool [has_hyperparam](#) (const std::string &name) const

- double [get_hyperparam_value](#) (const std::string &name) const
- int [get_hyperparam_index](#) (const std::string &name) const
- void [set_hyperparam_value](#) (const std::string &name, double value)
- void [rename_hyperparam](#) (const std::string &old_name, const std::string &new_name)
- [StringVecT](#) [get_param_names](#) () const
- void [set_param_names](#) (const [StringVecT](#) &desc)
- [StringVecT](#) [get_hyperparam_names](#) () const
- void [set_hyperparam_names](#) (const [StringVecT](#) &desc)
- template<class RngT >
 [ParamT](#) [sample_prior](#) (RngT &rng)
- [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_↵](#), double kappa=[default_intensity_kappa](#))
- static prior_hessian::TruncatedParetoDist [make_prior_component_sigma](#) (double [min_sigma](#), double max_[↵](#)sigma, double alpha=[default_alpha_sigma](#))
- static void [set_rng_seed](#) (RngSeedT seed)
- static [ParallelRngManagerT](#) & [get_rng_manager](#) ()
- static [ParallelRngGeneratorT](#) & [get_rng_generator](#) ()
- static void [check_size](#) (const [ImageSizeT](#) &size_)

Check the size argument for the model.

Static Public Attributes

- static const std::string [DefaultSeperableInitEstimator](#) = "TrustRegion"
- static const double [bounds_epsilon](#) = 1.0E-6
- static const double [global_min_psf_sigma](#) = 1E-1
- static const double [global_max_psf_sigma](#) = 1E2
- static const double [default_beta_pos](#) = 3
- static const double [default_sigma_pos](#) = 1
- static const double [default_mean_l](#) = 300
- static const double [default_max_l](#) = INFINITY
- static const double [default_intensity_kappa](#) = 2
- static const double [default_pixel_mean_bg](#) = 4
- static const double [default_alpha_sigma](#) = 2
- static const [ImageCoordT](#) num_dim =2
- static const [ImageCoordT](#) global_min_size =3
- static const [ImageCoordT](#) global_max_size =512

Protected Member Functions

- void [update_internal_1D_estimators](#) ()

Static Protected Member Functions

- static double [compute_max_sigma_ratio](#) (const [VecT](#) &min_sigma, const [VecT](#) &max_sigma)

Protected Attributes

- double [mcmc_candidate_eta_y](#)
- double [mcmc_candidate_eta_sigma](#)
- [VecT](#) min_sigma
- [Gauss1DsMAP](#) x_model
- [Gauss1DsMAP](#) y_model
- CompositeDist prior
- [IdxT](#) num_params
- [IdxT](#) num_hyperparams
- [ParamT](#) lbound
- [ParamT](#) ubound
- [ImageSizeT](#) size

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::Gauss2DsxModel::Gauss2DsxModel (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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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 166 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 175 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 184 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 196 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::Gauss2DsxyModel::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::Gauss2DsxModel::get_max_sigma () const [inline]`

Definition at line 127 of file Gauss2DsxModel.h.

References `get_max_sigma_ratio()`, and `get_min_sigma()`.

8.19.4.19 `double mappel::Gauss2DsxModel::get_max_sigma (IdxT dim) const [inline]`

Definition at line 131 of file Gauss2DsxModel.h.

References `get_max_sigma_ratio()`, and `get_min_sigma()`.

8.19.4.20 `double mappel::Gauss2DsxModel::get_max_sigma_ratio () const [inline]`

Definition at line 135 of file Gauss2DsxModel.h.

References `mappel::PointEmitterModel::get_ubound()`.

Referenced by `get_max_sigma()`.

8.19.4.21 `VecT mappel::Gauss2DsxModel::get_min_sigma () const [inline]`

Definition at line 122 of file Gauss2DsxModel.h.

References `min_sigma`.

Referenced by `get_max_sigma()`.

8.19.4.22 `double mappel::Gauss2DsxModel::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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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`.

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 97 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::Gauss2Ds↵`
`Model::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 91 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::Gauss2Ds↵`
`Model::make_default_prior_beta_position()`, `mappel::Gauss1DsModel::make_prior_beta_position()`, `mappel::Gauss2↵`
`DModel::make_prior_beta_position()`, `mappel::Gauss1DModel::make_prior_beta_position()`, and `mappel::Gauss2Ds↵`
`Model::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 84 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::Gauss2↵`
`DsModel::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()`.

8.19.4.55 `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 103 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 Gauss2DsxyModel.h.

References `mappel::Gauss2DsxyModel::Stencil::Stencil()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

8.19.4.58 `void mappel::Gauss2DsxyModel::pixel_grad (int i, int j, const Stencil & s, ParamT & pgrad) const [inline]`

Definition at line 178 of file Gauss2DsxyModel.h.

References `mappel::Gauss2DsxyModel::Stencil::DX`, `mappel::Gauss2DsxyModel::Stencil::DXS`, `mappel::Gauss2DsxyModel::Stencil::DY`, `mappel::Gauss2DsxyModel::Stencil::DYS`, `mappel::Gauss2DsxyModel::Stencil::l()`, `mappel::Gauss2DsxyModel::Stencil::X`, and `mappel::Gauss2DsxyModel::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::l()`, `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::l()`, `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::l()`, `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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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::Gauss2DsxyModel::sample_mcmc_candidate (int sample_index, ParamT & canidate_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 212 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 225 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 148 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 109 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 236 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 256 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::make_stencil(), mappel::Gauss1DsModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), make_stencil(), mappel::Gauss2DsModel::make_stencil(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.19.4.85 **BoolVecT** mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & *theta*) const [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

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 emitter intensity mean

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::Gauss2DsModel::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::Gauss2DsMA←
P(), mappel::Gauss2DsMLE::Gauss2DsMLE(), mappel::Gauss2DsModel::Gauss2DsModel(), mappel::Image←
Format2DBase::get_num_pixels(), mappel::ImageFormat2DBase::get_size(), mappel::ImageFormat2DBase::get←
_stats(), mappel::methods::likelihood::grad(), mappel::methods::likelihood::grad2(), mappel::methods::likelihood←
::debug::grad_components(), mappel::methods::likelihood::hessian(), mappel::methods::likelihood::debug::hessian←
_components(), mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(),
mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh_components(), mappel::Gauss2DModel←
::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::←
Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal←
_position(), mappel::ImageFormat2DBase::make_image(), mappel::ImageFormat2DBase::make_image_stack(),
mappel::Gauss2DModel::make_internal_1Dsum_estimator(), mappel::Gauss2DsModel::make_internal_1Dsum_←
estimator(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_←
position(), mappel::Gauss2DsModel::make_prior_normal_position(), mappel::Gauss2DsModel::make_prior_normal←
_position(), mappel::methods::model_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods←
::likelihood::rllh(), mappel::methods::likelihood::debug::rllh_components(), mappel::ImageFormat2DBase::set←
size(), mappel::Gauss2DModel::set_size(), mappel::Gauss2DsModel::set_size(), mappel::methods::simulate_image(),
mappel::methods::simulate_image_from_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2Ds←
Model::Stencil::Stencil(), mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2Ds←
Model::update_internal_1Dsum_estimators().

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::Point←
EmitterModel::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.

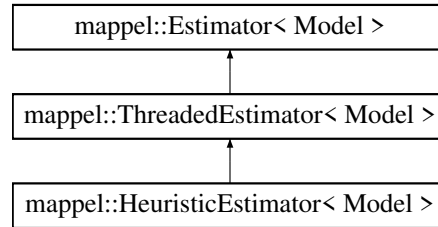
The documentation for this class was generated from the following file:

- [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 Member Functions

- [HeuristicEstimator](#) (Model &model)
- `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)`
- `StatsT get_stats ()`
- `StatsT get_debug_stats ()`
- `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)`

Protected Member Functions

- `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.

8.20.1 Detailed Description

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

Definition at line 133 of file estimator.h.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 `template<class Model > mappel::HeuristicEstimator< Model >::HeuristicEstimator (Model & model)`
[\[inline\]](#)

Definition at line 135 of file estimator.h.

8.20.3 Member Function Documentation

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

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

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

Definition at line 287 of file estimator_impl.h.

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

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

8.20.3.2 `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 142 of file estimator_impl.h.

References [mappel::methods::observed_information\(\)](#).

8.20.3.3 `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) [inline], [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 192 of file estimator_impl.h.

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

8.20.3.4 `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 151 of file estimator_impl.h.

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

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

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

Definition at line 59 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

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

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

Definition at line 77 of file estimator_impl.h.

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

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

Definition at line 68 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

8.20.3.8 `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 85 of file estimator_impl.h.

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

Definition at line 97 of file estimator_impl.h.

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

8.20.3.10 `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 106 of file estimator_impl.h.

8.20.3.11 `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 117 of file estimator_impl.h.

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

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

8.20.3.12 `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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

8.20.3.13 `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 222 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

8.20.3.14 `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 246 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::num_threads](#), [mappel::Estimator< Model >::record_walltime\(\)](#), [omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::rethrow\(\)](#), and [omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::run\(\)](#).

8.20.3.15 `template<class Model > StatsT mappel::ThreadedEstimator< Model >::get_debug_stats () [virtual], [inherited]`

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

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

Definition at line 281 of file estimator_impl.h.

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

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

Definition at line 45 of file estimator_impl.h.

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

8.20.3.17 `template<class Model > StatsT mappel::ThreadedEstimator< Model >::get_stats () [virtual], [inherited]`

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

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

Definition at line 271 of file estimator_impl.h.

References [mappel::Estimator< Model >::get_stats\(\)](#), [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::CGaussMLE< Model >::get_stats\(\)](#), and [mappel::IterativeMaximizer< Model >::get_stats\(\)](#).

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

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

Definition at line 137 of file estimator.h.

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

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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

8.20.4 Member Data Documentation

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

Definition at line 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by mappel::SimulatedAnnealingMaximizer< Model >::anneal(), mappel::IterativeMaximizer< Model >::backtrack(), mappel::ThreadedEstimator< Model >::clear_stats(), mappel::CGaussMLE< Model >::compute_estimate(), mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate(), mappel::IterativeMaximizer< Model >::compute_estimate(), mappel::CGaussMLE< Model >::compute_estimate_debug(), mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug(), mappel::IterativeMaximizer< Model >::compute_estimate_debug(), mappel::IterativeMaximizer< Model >::compute_profile_estimate(), mappel::ThreadedEstimator< Model >::estimate_max_stack(), mappel::ThreadedEstimator< Model >::estimate_profile_stack(), mappel::IterativeMaximizer< Model >::local_maximize(), mappel::NewtonDiagonalMaximizer< Model >::maximize(), mappel::NewtonMaximizer< Model >::maximize(), mappel::QuasiNewtonMaximizer< Model >::maximize(), and mappel::TrustRegionMaximizer< Model >::maximize().

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

Referenced by mappel::ThreadedEstimator< Model >::get_stats(), and mappel::IterativeMaximizer< Model >::get_stats().

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

Definition at line 128 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()`.

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

Definition at line 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

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

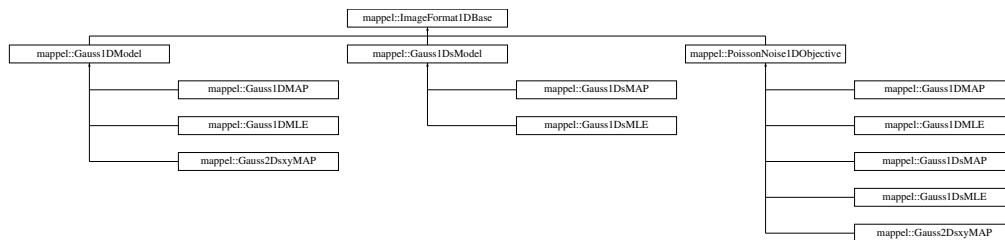
- [estimator.h](#)
- [estimator_impl.h](#)

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:

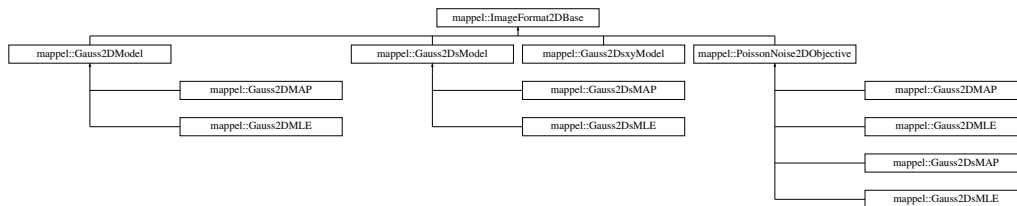
- [ImageFormat1DBase.h](#)
- [ImageFormat1DBase.cpp](#)

8.22 mappel::ImageFormat2DBase Class Reference

A virtual base class for 2D image localization objectives.

```
#include </home/travis/build/markjolah/Mappel/include/Mappel/ImageFormat2DBase.h>
```

Inheritance diagram for mappel::ImageFormat2DBase:



Public Types

- using [ImageCoordT](#) = uint32_t
- using [ImagePixelT](#) = double
- template<class CoordT >
using [ImageSizeShapeT](#) = arma::Col< CoordT >
- template<class CoordT >
using [ImageSizeVecShapeT](#) = arma::Mat< CoordT >
- using [ImageSizeT](#) = [ImageSizeShapeT](#)< [ImageCoordT](#) >
- using [ImageSizeVecT](#) = [ImageSizeVecShapeT](#)< [ImageCoordT](#) >
- template<class PixelT >
using [ImageShapeT](#) = arma::Mat< PixelT >
- template<class PixelT >
using [ImageStackShapeT](#) = arma::Cube< PixelT >
- using [ImageT](#) = [ImageShapeT](#)< [ImagePixelT](#) >
- using [ImageStackT](#) = [ImageStackShapeT](#)< [ImagePixelT](#) >

Public Member Functions

- [StatsT get_stats](#) () const
- [ImageT make_image](#) () const
- [ImageStackT make_image_stack](#) ([ImageCoordT](#) n) const
- [ImageCoordT get_size_image_stack](#) (const [ImageStackT](#) &stack) const
- [ImageT get_image_from_stack](#) (const [ImageStackT](#) &stack, [ImageCoordT](#) n) const
- template<class ImT >
void [set_image_in_stack](#) ([ImageStackT](#) &stack, [ImageCoordT](#) n, const ImT &im) const
- const [ImageSizeT](#) & [get_size](#) () const
- [ImageCoordT get_size](#) ([IdxT](#) idx) const
- [ImageCoordT get_num_pixels](#) () const
- void [set_size](#) (const [ImageSizeT](#) &size_)
- void [check_image_shape](#) (const [ImageT](#) &im) const
Check the shape of a single images is correct for model size.
- void [check_image_shape](#) (const [ImageStackT](#) &ims) const
Check the shape of a stack of images is correct for model size.

Static Public Member Functions

- static void [check_size](#) (const [ImageSizeT](#) &size_)
Check the size argument for the model.

Static Public Attributes

- static const [ImageCoordT](#) [num_dim](#) =2
- static const [ImageCoordT](#) [global_min_size](#) =3
- static const [ImageCoordT](#) [global_max_size](#) =512

Protected Member Functions

- [ImageFormat2DBase](#) ()=default
- [ImageFormat2DBase](#) (const [ImageSizeT](#) &size)
- [ImageFormat2DBase](#) (const [ImageFormat2DBase](#) &)
- [ImageFormat2DBase](#) ([ImageFormat2DBase](#) &&)
- [ImageFormat2DBase](#) & [operator=](#) (const [ImageFormat2DBase](#) &)
- [ImageFormat2DBase](#) & [operator=](#) ([ImageFormat2DBase](#) &&)

Protected Attributes

- [ImageSizeT](#) [size](#)

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().

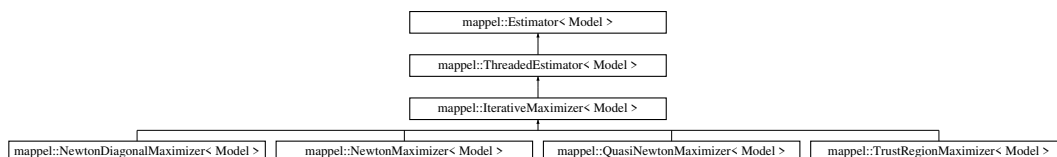
The documentation for this class was generated from the following files:

- [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` = 99, `ExitCode::MaxIter` = 6, `ExitCode::MaxBacktracks` = 5, `ExitCode::TrustRegionRadius` = 4,
`ExitCode::GradRatio` = 3, `ExitCode::FunctionChange` = 2, `ExitCode::StepSize` = 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` = 7

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)
 - 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.*
- void [record_walltime](#) (ClockT::time_point start_walltime, int nimages)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = sqrt(std::numeric_limits<double>::epsilon())
- double [delta](#) = sqrt(std::numeric_limits<double>::epsilon())
- 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](#) [exit_counts](#)
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.

8.23.1 Detailed Description

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

Definition at line 195 of file estimator.h.

8.23.2 Member Enumeration Documentation

8.23.2.1 `template<class Model > enum mappel::IterativeMaximizer::ExitCode : IdxT [strong]`

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Error

Definition at line 198 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 356 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 495 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_parameter_scalar`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::IterativeMaximizer< Model >::MaxBacktracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `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 469 of file `estimator_impl.h`.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::exit_counts`, `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 142 of file `estimator_impl.h`.

References `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 593 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::Error`, `mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence()`, `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::print_image()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `mappel::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::theta()`.

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 618 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::Error`, `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::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `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 637 of file `estimator_impl.h`.

References [mappel::IterativeMaximizer< Model>::Error](#), [mappel::IterativeMaximizer< Model>::maximize\(\)](#), [mappel::Estimator< Model>::model](#), [mappel::IterativeMaximizer< Model>::MaximizerData::record_exit\(\)](#), [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 571 of file `estimator_impl.h`.

References [mappel::IterativeMaximizer< Model>::delta](#), [mappel::IterativeMaximizer< Model>::epsilon](#), [mappel::IterativeMaximizer< Model>::MaximizerData::fixed_parameter_scalar](#), [mappel::IterativeMaximizer< Model>::FunctionChange](#), [mappel::IterativeMaximizer< Model>::MaximizerData::grad](#), [mappel::IterativeMaximizer< Model>::MaximizerData::has_fixed_parameters](#), [mappel::IterativeMaximizer< Model>::MaximizerData::record_exit\(\)](#), [mappel::IterativeMaximizer< Model>::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model>::MaximizerData::saved_theta\(\)](#), [mappel::IterativeMaximizer< Model>::StepSize](#), and [mappel::IterativeMaximizer< Model>::MaximizerData::theta\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model>::backtrack\(\)](#), [mappel::NewtonDiagonalMaximizer< Model>::maximize\(\)](#), [mappel::NewtonMaximizer< Model>::maximize\(\)](#), and [mappel::QuasiNewtonMaximizer< Model>::maximize\(\)](#).

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

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

References [mappel::methods::estimate_max\(\)](#).

Referenced by [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 77 of file `estimator_impl.h`.

References [mappel::methods::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 68 of file `estimator_impl.h`.

References [mappel::methods::estimate_max\(\)](#).

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 85 of file estimator_impl.h.

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 97 of file estimator_impl.h.

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

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 106 of file estimator_impl.h.

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 117 of file estimator_impl.h.

References `mappel::methods::observed_information()`, 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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

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 222 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

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 246 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::num_threads](#), [mappel::Estimator< Model >::record_walltime\(\)](#), [omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow\(\)](#), and [omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run\(\)](#).

8.23.4.18 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_debug_stats () [virtual]`

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

Definition at line 453 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 45 of file estimator_impl.h.

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 424 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::IterativeMaximizer< Model >::Error](#), [mappel::IterativeMaximizer< Model >::exit_counts](#), [mappel::IterativeMaximizer< Model >::FunctionChange](#), [mappel::ThreadedEstimator< Model >::get_stats\(\)](#), [mappel::IterativeMaximizer< Model >::GradRatio](#), [mappel::IterativeMaximizer< Model >::max_backtracks](#), [mappel::IterativeMaximizer< Model >::max_iterations](#), [mappel::IterativeMaximizer< Model >::MaxBacktracks](#), [mappel::IterativeMaximizer< Model >::MaxIter](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::Estimator< Model >::num_estimations](#), [mappel::IterativeMaximizer< Model >::StepSize](#), [mappel::IterativeMaximizer< Model >::total_backtracks](#), [mappel::IterativeMaximizer< Model >::total_der_evals](#), [mappel::IterativeMaximizer< Model >::total_fun_evals](#), [mappel::IterativeMaximizer< Model >::total_iterations](#), and [mappel::IterativeMaximizer< Model >::TrustRegionRadius](#).

Referenced by [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 & rlh)`

Perform a local maximization to finish off a simulated annealing run.

Definition at line 655 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::rlh`, 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 >::compute_profile_estimate()`, and `mappel::Estimator< Model >::~~Estimator()`.

8.23.4.28 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const MaximizerData & data) [protected]`

Definition at line 482 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::exit_code`, `mappel::IterativeMaximizer< Model >::exit_counts`, `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()`.

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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 232 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 = sqrt(std::numeric_limits<double>::epsilon())` [protected]

Definition at line 229 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< 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 = sqrt(std::numeric_limits<double>::epsilon())` [protected]

Definition at line 228 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`.

8.23.5.4 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::exit_counts` [protected]

Definition at line 240 of file estimator.h.

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

8.23.5.5 `template<class Model> double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` [protected]

Definition at line 231 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` [protected]

Definition at line 242 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 233 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 225 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `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 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::maximize()`.

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 128 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::IterativeMaximizer< Model >::NumExitCodes = 7` [static]

Definition at line 197 of file estimator.h.

8.23.5.15 `template<class Model > int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected]

Definition at line 237 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 239 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 238 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 236 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 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

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

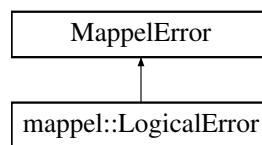
- [estimator.h](#)
- [estimator_impl.h](#)

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:

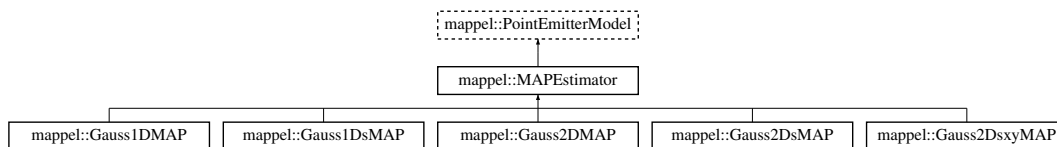
- [util.h](#)

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_↵, double 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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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

Definition at line 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.25.4.20 `StatsT mappel::PointEmitterModel::get_stats () const` `[inherited]`

Definition at line 124 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::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().

8.25.4.22 `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.

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 97 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 91 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 84 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 103 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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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 212 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 225 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 157 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 148 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 109 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.25.4.51 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 236 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 256 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::Gauss2DsModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil_xy()`, 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 296 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::Gauss2DModel::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 emitter intensity mean

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::MCMCAaptor1D::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_exit](#) ([ExitCode](#) code)
- void [record_iteration](#) ()
Record an iteration point (derivatives computed) Using the saved theta as the default.
- 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](#)
- [ExitCode](#) [exit_code](#) =[ExitCode::Unassigned](#)
- [VecT](#) [fixed_parameter_scalar](#)
- bool [has_fixed_parameters](#) =false

Protected Attributes

- [StencilT](#)< Model > [s0](#)
- [StencilT](#)< Model > [s1](#)
- bool [current_stencil](#)
- [ParamVecT](#)< Model > [theta_seq](#)
- [VecT](#) [seq_rllh](#)
- [IdxVecT](#) [backtrack_idx](#)
- 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 244 of file estimator.h.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 `template<class Model > mappel::IterativeMaximizer< Model >::MaximizerData::MaximizerData (const Model & model, const ModelDataT< Model > & im, const StencilT< Model > & s, bool save_seq = false, int max_seq_len = 0)`

Definition at line 363 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::backtrack_idx`, `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 270 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 269 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::compute_estimate()`, and `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 271 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 291 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 264 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 398 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_exit (ExitCode code)`

Definition at line 411 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::exit_code`.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, and `mappel::IterativeMaximizer< Model >::convergence_test()`.

8.26.3.8 `template<class Model > void mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration ()`
[inline]

Record an iteration point (derivatives computed) Using the saved theta as the default.

Definition at line 260 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()`.

8.26.3.9 `template<class Model > void mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration (const ParamT< Model > & accepted_theta)`

Record an iteration point (derivatives computed)

Definition at line 386 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`.

8.26.3.10 `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 284 of file estimator.h.

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

8.26.3.11 `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 279 of file estimator.h.

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

8.26.3.12 `template<class Model > StencilT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::saved_stencil () [inline]`

Get the saved stencil.

Definition at line 286 of file estimator.h.

8.26.3.13 `template<class Model > ParamT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta () [inline]`

Get the saved stencil's theta.

Definition at line 290 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, and `mappel::IterativeMaximizer< Model >::convergence_test()`.

8.26.3.14 `template<class Model > void mappel::IterativeMaximizer< Model >::MaximizerData::set_fixed_parameters (const IdxVecT & fixed_parameters)`

Definition at line 417 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_parameter_scalar`, and `mappel::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters`.

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

8.26.3.15 `template<class Model > void mappel::IterativeMaximizer< Model >::MaximizerData::set_stencil (const StencilT< Model > & s) [inline]`

Definition at line 274 of file estimator.h.

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

8.26.3.16 `template<class Model > StencilT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::stencil () [inline]`

Get the current stencil.

Definition at line 273 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()`.

8.26.3.17 `template<class Model > ParamT<Model>& mappel::IterativeMaximizer< Model >::MaximizerData::theta () [inline]`

Get the current stencil's theta.

Definition at line 288 of file estimator.h.

Referenced by `mappel::TrustRegionMaximizer< Model >::compute_bound_scaling_vec()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `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 302 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 298 of file estimator.h.

8.26.4.3 `template<class Model> ExitCode mappel::IterativeMaximizer< Model>::MaximizerData::exit_code`
`=ExitCode::Unassigned`

Definition at line 254 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::MaximizerData::record_exit()`, and `mappel::IterativeMaximizer< Model>::record_run_statistics()`.

8.26.4.4 `template<class Model> VecT mappel::IterativeMaximizer< Model>::MaximizerData::fixed_parameter_scalar`

Definition at line 293 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 247 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::backtrack()`, and `mappel::IterativeMaximizer< Model>::convergence_test()`.

8.26.4.6 `template<class Model> bool mappel::IterativeMaximizer< Model>::MaximizerData::has_fixed_parameters =false`

Definition at line 294 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model>::backtrack()`, `mappel::IterativeMaximizer< Model>::convergence_test()`, and `mappel::IterativeMaximizer< Model>::MaximizerData::set_fixed_parameters()`.

8.26.4.7 `template<class Model> const ModelDataT<Model>& mappel::IterativeMaximizer< Model>::MaximizerData::im`

Definition at line 246 of file estimator.h.

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

8.26.4.8 `template<class Model > VecT mappel::IterativeMaximizer< Model >::MaximizerData::lbound`

Definition at line 249 of file estimator.h.

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

8.26.4.9 `template<class Model > const int mappel::IterativeMaximizer< Model >::MaximizerData::max_seq_len`
[protected]

Definition at line 304 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`.

8.26.4.10 `template<class Model > int mappel::IterativeMaximizer< Model >::MaximizerData::nBacktracks =0`

Definition at line 251 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.26.4.11 `template<class Model > int mappel::IterativeMaximizer< Model >::MaximizerData::nIterations =0`

Definition at line 252 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::MaximizerData::record_iteration()`, and `mappel::IterativeMaximizer< Model >::record_run_statistics()`.

8.26.4.12 `template<class Model > double mappel::IterativeMaximizer< Model >::MaximizerData::rllh`

Definition at line 250 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 297 of file estimator.h.

8.26.4.14 `template<class Model > StencilT<Model> mappel::IterativeMaximizer< Model >::MaximizerData::s1`
[protected]

Definition at line 297 of file estimator.h.

8.26.4.15 `template<class Model> bool mappel::IterativeMaximizer< Model>::MaximizerData::save_seq`

Definition at line 253 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 303 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 301 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 248 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 300 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 249 of file estimator.h.

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

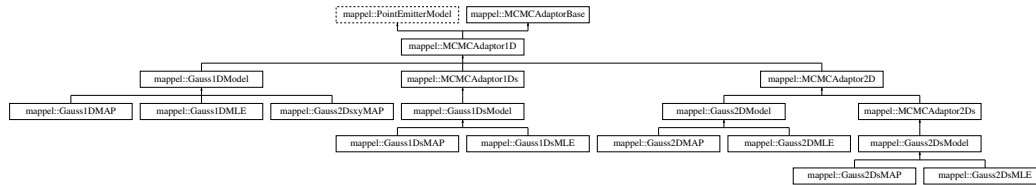
The documentation for this class was generated from the following files:

- [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 247 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 264 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), mappel::PointEmitterModel::lbound, mappel::Point↔EmitterModel::num_params, and mappel::PointEmitterModel::ubound.

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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

Definition at line 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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.

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 97 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 91 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 84 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()`.

8.27.4.36 `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 103 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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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 212 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 225 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 157 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()`.

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

Definition at line 148 of file PointEmitterModel.cpp.

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

8.27.4.57 `void mappel::PointEmitterModel::set_rng_seed (RngSeedT seed)` `[static],[inherited]`

Definition at line 109 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.27.4.58 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)` `[inherited]`

Definition at line 236 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()`.

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

Definition at line 256 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 296 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 emitter intensity mean

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().

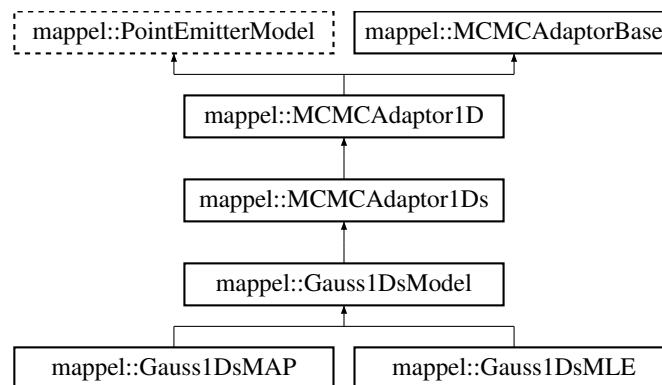
The documentation for this class was generated from the following files:

- [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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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

Definition at line 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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::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().

8.28.4.24 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::Gauss2DsxModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

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`.

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_I, double kappa = default_intensity_kappa)` `[static]`, `[inherited]`

Definition at line 97 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()`.

8.28.4.34 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` `[static]`, `[inherited]`

Definition at line 91 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()`.

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

Definition at line 84 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()`.

8.28.4.36 `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 103 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 275 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 316 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), 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 212 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_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()`.

8.28.4.51 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 225 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 157 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 148 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 109 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.28.4.58 void mappel::PointEmitterModel::set_ubound (const ParamT & *ubound*) [inherited]

Definition at line 236 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 256 of file PointEmitterModel.cpp.

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

Referenced by mappel::Gauss2DModel::initial_theta_estimate(), mappel::Gauss2DsModel::initial_theta_estimate(), mappel::Gauss1DModel::make_stencil(), mappel::Gauss1DsModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DModel::make_stencil(), mappel::Gauss2DsxyModel::make_stencil(), mappel::Gauss2DsModel::make_stencil(), and mappel::PointEmitterModel::theta_stack_in_bounds().

8.28.4.60 BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & *theta*) const [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

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 emitter intensity mean

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().

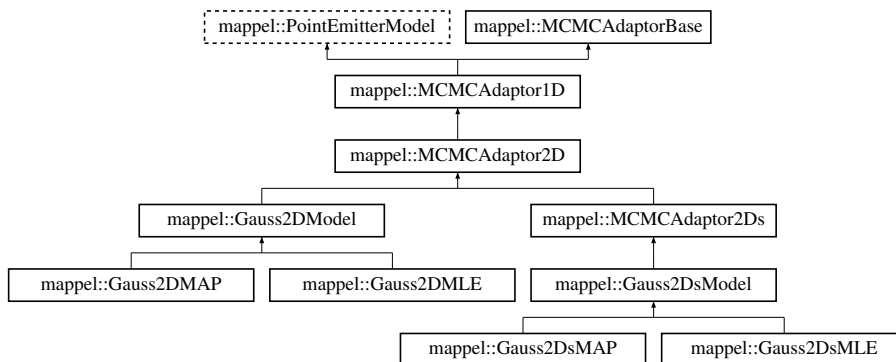
The documentation for this class was generated from the following files:

- [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_↵ 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

- [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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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

Definition at line 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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`.

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 97 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 91 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()`.

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

Definition at line 84 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()`.

8.29.4.36 `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 103 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 275 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 316 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), 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 212 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_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()`.

8.29.4.51 `void mappel::PointEmitterModel::set_lbound (const ParamT & lbound)` `[inherited]`

Definition at line 225 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 157 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 148 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 109 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.29.4.58 void mappel::PointEmitterModel::set_ubound (const ParamT & *ubound*) [inherited]

Definition at line 236 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 256 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 296 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 emitter intensity mean

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::D()`, `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::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::PointEmitterModel::operator=()`, `mappel::PointEmitterModel::reflected_theta()`, `mappel::PointEmitterModel::set_bounds()`, `mappel::PointEmitterModel::set_lbound()`, `mappel::PointEmitterModel::set_prior()`, `mappel::PointEmitterModel::set_ubound()`, and `mappel::PointEmitterModel::theta_in_bounds()`.

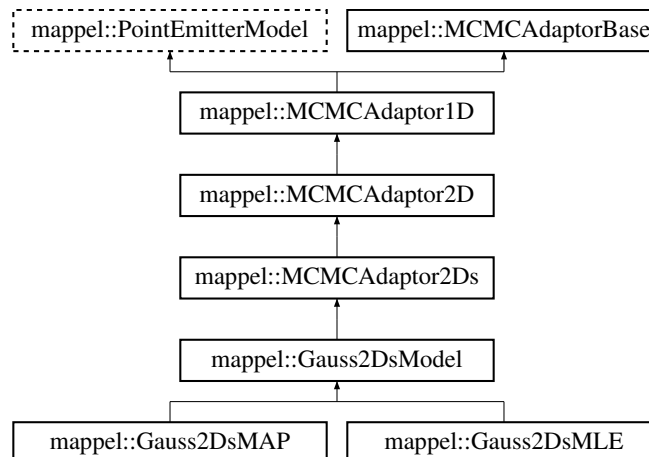
The documentation for this class was generated from the following files:

- [MCMCAdaptor2D.h](#)
- [MCMCAdaptor2D.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by mappel::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 306 of file PointEmitterModel.cpp.

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

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

Definition at line 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 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::Gauss2DsxModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`, `mappel::Gauss1DModel::pixel_hess_update()`, `mappel::Gauss1DsModel::pixel_hess_update()`, `mappel::Gauss2DModel::pixel_hess_update()`, and `mappel::Gauss2DsModel::pixel_hess_update()`.

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`.

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 97 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 91 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 84 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 103 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 275 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 316 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), 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 212 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 225 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 157 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 148 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 109 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.30.4.58 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 236 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 256 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::Gauss2DsModel::make_stencil()`, `mappel::Gauss2DsModel::make_stencil_xy()`, 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 296 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 emitter intensity mean

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().

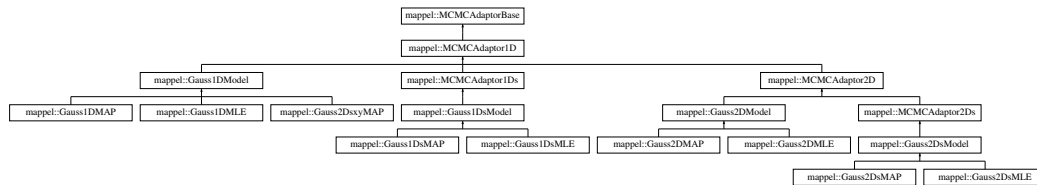
The documentation for this class was generated from the following files:

- [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:

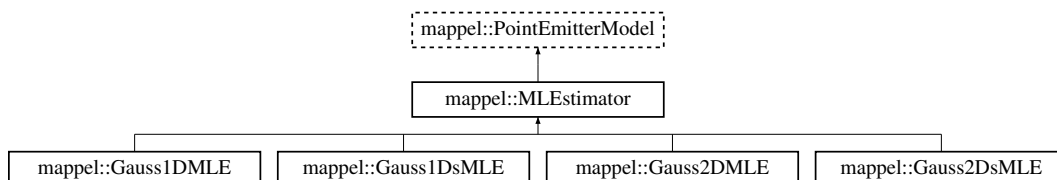
- [MCMCAdaptorBase.h](#)
- [MCMCAdaptorBase.cpp](#)

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 247 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 264 of file PointEmitterModel.cpp.

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

Referenced by `mappel::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 306 of file PointEmitterModel.cpp.

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

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

Definition at line 166 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 175 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 184 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 196 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 119 of file PointEmitterModel.cpp.

References mappel::rng_manager.

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

Definition at line 114 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.32.4.20 `StatsT mappel::PointEmitterModel::get_stats () const` `[inherited]`

Definition at line 124 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::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().

8.32.4.22 `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`.

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 97 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()`.

8.32.4.32 `prior_hessian::ScaledSymmetricBetaDist mappel::PointEmitterModel::make_prior_component_position_beta (IdxT size, double pos_beta = default_beta_pos)` `[static]`, `[inherited]`

Definition at line 91 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()`.

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

Definition at line 84 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()`.

8.32.4.34 `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 103 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 275 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 316 of file PointEmitterModel.cpp.

References `mappel::PointEmitterModel::check_param_shape()`, 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 212 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 225 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 157 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 148 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 109 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.32.4.51 `void mappel::PointEmitterModel::set_ubound (const ParamT & ubound) [inherited]`

Definition at line 236 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 256 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().

8.32.4.53 BoolVecT mappel::PointEmitterModel::theta_stack_in_bounds (const ParamVecT & *theta*) const [inherited]

Definition at line 296 of file PointEmitterModel.cpp.

References mappel::PointEmitterModel::check_param_shape(), and mappel::PointEmitterModel::theta_in_bounds().

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::Gauss2↔DsModel::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 emitter intensity mean

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::MCMCAaptor1D::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:

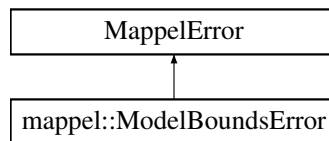
- [MLEstimator.h](#)

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.

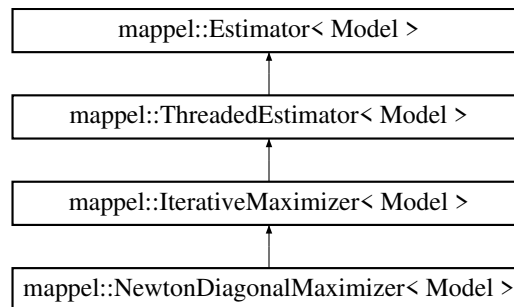
The documentation for this struct was generated from the following file:

- [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` = 99, `ExitCode::MaxIter` = 6, `ExitCode::MaxBacktracks` = 5, `ExitCode::TrustRegionRadius` = 4,
`ExitCode::GradRatio` = 3, `ExitCode::FunctionChange` = 2, `ExitCode::StepSize` = 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](#) = 7

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_walltime](#) (ClockT::time_point start_walltime, int nimages)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = sqrt(std::numeric_limits<double>::epsilon())
- double [delta](#) = sqrt(std::numeric_limits<double>::epsilon())
- 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](#) [exit_counts](#)
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.

8.34.1 Detailed Description

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

Definition at line 322 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 324 of file estimator.h.

8.34.3 Member Enumeration Documentation

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

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Error

Definition at line 198 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 326 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 495 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_parameter_scalar`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::IterativeMaximizer< Model >::MaxBacktracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `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 469 of file `estimator_impl.h`.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::exit_counts`, `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 142 of file `estimator_impl.h`.

References `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 593 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::Error](#), [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence\(\)](#), [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::record_exit\(\)](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

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 618 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::Error](#), [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::IterativeMaximizer< Model >::MaximizerData::record_exit\(\)](#), [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 637 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::Error](#), [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::IterativeMaximizer< Model >::MaximizerData::record_exit\(\)](#), [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 571 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_parameter_scalar`, `mappel::IterativeMaximizer< Model >::FunctionChange`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::IterativeMaximizer< Model >::StepSize`, and `mappel::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

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

Definition at line 59 of file estimator_impl.h.

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

Referenced by `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 77 of file estimator_impl.h.

References `mappel::methods::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 68 of file estimator_impl.h.

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

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 85 of file estimator_impl.h.

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 97 of file estimator_impl.h.

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

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 106 of file estimator_impl.h.

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 117 of file estimator_impl.h.

References `mappel::methods::observed_information()`, 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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

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 222 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

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 246 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

8.34.5.18 `template<class Model> StatsT mappel::IterativeMaximizer< Model>::get_debug_stats () [virtual],
[inherited]`

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

Definition at line 453 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 45 of file estimator_impl.h.

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 424 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model>::delta](#), [mappel::IterativeMaximizer< Model>::epsilon](#), [mappel::IterativeMaximizer< Model>::Error](#), [mappel::IterativeMaximizer< Model>::exit_counts](#), [mappel::IterativeMaximizer< Model>::FunctionChange](#), [mappel::ThreadedEstimator< Model>::get_stats\(\)](#), [mappel::IterativeMaximizer< Model>::GradRatio](#), [mappel::IterativeMaximizer< Model>::max_backtracks](#), [mappel::IterativeMaximizer< Model>::max_←_iterations](#), [mappel::IterativeMaximizer< Model>::MaxBacktracks](#), [mappel::IterativeMaximizer< Model>::MaxIter](#), [mappel::ThreadedEstimator< Model>::mtx](#), [mappel::Estimator< Model>::num_estimations](#), [mappel::Iterative←Maximizer< Model>::StepSize](#), [mappel::IterativeMaximizer< Model>::total_backtracks](#), [mappel::IterativeMaximizer< Model>::total_der_evals](#), [mappel::IterativeMaximizer< Model>::total_fun_evals](#), [mappel::IterativeMaximizer< Model>::total_iterations](#), and [mappel::IterativeMaximizer< Model>::TrustRegionRadius](#).

Referenced by [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 655 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 664 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::methods::objective::grad2()`, `mappel::IterativeMaximizer< Model >::max_iterations`, and `mappel::Estimator< Model >::model`.

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 329 of file `estimator.h`.

8.34.5.29 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const MaximizerData & data)`
`[protected], [inherited]`

Definition at line 482 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::MaximizerData::exit_code`, `mappel::IterativeMaximizer< Model >::exit_counts`, `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()`.

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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 232 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 = sqrt(std::numeric_limits<double>::epsilon())` [protected], [inherited]

Definition at line 229 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< 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 = sqrt(std::numeric_limits<double>::epsilon())` [protected], [inherited]

Definition at line 228 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`.

8.34.6.4 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::exit_counts` [protected], [inherited]

Definition at line 240 of file estimator.h.

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

8.34.6.5 `template<class Model> double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` [protected],
[inherited]

Definition at line 231 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` [protected],
[inherited]

Definition at line 242 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 233 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 225 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `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 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::maximize()`.

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 128 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::IterativeMaximizer< Model >::NumExitCodes = 7` [static],
[inherited]

Definition at line 197 of file estimator.h.

8.34.6.15 `template<class Model > int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected],
[inherited]

Definition at line 237 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 239 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 238 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 236 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 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

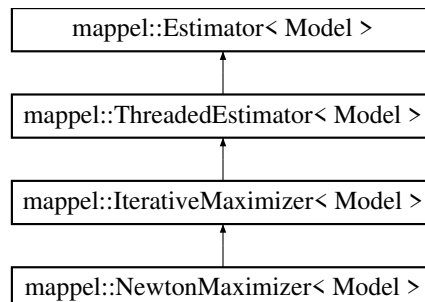
The documentation for this class was generated from the following files:

- [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` = 99, `ExitCode::MaxIter` = 6, `ExitCode::MaxBacktracks` = 5, `ExitCode::TrustRegionRadius` = 4,
`ExitCode::GradRatio` = 3, `ExitCode::FunctionChange` = 2, `ExitCode::StepSize` = 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`, `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` = 7

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_walltime](#) (ClockT::time_point start_walltime, int nimages)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = sqrt(std::numeric_limits<double>::epsilon())
- double [delta](#) = sqrt(std::numeric_limits<double>::epsilon())
- 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](#) [exit_counts](#)
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.

8.35.1 Detailed Description

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

Definition at line 336 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 338 of file estimator.h.

8.35.3 Member Enumeration Documentation

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

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Error

Definition at line 198 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 340 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 495 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_parameter_scalar`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::IterativeMaximizer< Model >::MaxBacktracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `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 469 of file `estimator_impl.h`.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::exit_counts`, `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 142 of file `estimator_impl.h`.

References `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 593 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::Error`, `mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence()`, `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::print_image()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `mappel::IterativeMaximizer< Model >::record_run_statistics()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::stencil()`, and `mappel::IterativeMaximizer< Model >::MaximizerData::theta()`.

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 618 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::Error`, `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::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `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 637 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::Error`, `mappel::IterativeMaximizer< Model >::maximize()`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `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 571 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_parameter_scalar`, `mappel::IterativeMaximizer< Model >::FunctionChange`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::IterativeMaximizer< Model >::StepSize`, and `mappel::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

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

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

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

Referenced by `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 77 of file `estimator_impl.h`.

References `mappel::methods::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 68 of file `estimator_impl.h`.

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

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 85 of file estimator_impl.h.

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 97 of file estimator_impl.h.

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

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 106 of file estimator_impl.h.

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 117 of file estimator_impl.h.

References `mappel::methods::observed_information()`, 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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

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 222 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::run()`.

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 246 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::num_threads](#), [mappel::Estimator< Model >::record_walltime\(\)](#), [omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::rethrow\(\)](#), and [omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::run\(\)](#).

8.35.5.18 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_debug_stats () [virtual], [inherited]`

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

Definition at line 453 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 45 of file estimator_impl.h.

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 424 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::IterativeMaximizer< Model >::Error](#), [mappel::IterativeMaximizer< Model >::exit_counts](#), [mappel::IterativeMaximizer< Model >::FunctionChange](#), [mappel::ThreadedEstimator< Model >::get_stats\(\)](#), [mappel::IterativeMaximizer< Model >::GradRatio](#), [mappel::IterativeMaximizer< Model >::max_backtracks](#), [mappel::IterativeMaximizer< Model >::max_iterations](#), [mappel::IterativeMaximizer< Model >::MaxBacktracks](#), [mappel::IterativeMaximizer< Model >::MaxIter](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::Estimator< Model >::num_estimations](#), [mappel::IterativeMaximizer< Model >::StepSize](#), [mappel::IterativeMaximizer< Model >::total_backtracks](#), [mappel::IterativeMaximizer< Model >::total_der_evals](#), [mappel::IterativeMaximizer< Model >::total_fun_evals](#), [mappel::IterativeMaximizer< Model >::total_iterations](#), and [mappel::IterativeMaximizer< Model >::TrustRegionRadius](#).

Referenced by [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 655 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 701 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::methods::objective::hessian()`, `mappel::IterativeMaximizer< Model >::max_iterations`, and `mappel::Estimator< Model >::model`.

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 343 of file estimator.h.

8.35.5.29 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const MaximizerData & data)` [protected],[inherited]

Definition at line 482 of file estimator_impl.h.

References mappel::IterativeMaximizer< Model >::MaximizerData::exit_code, mappel::IterativeMaximizer< Model >::exit_counts, 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().

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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 232 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 = sqrt(std::numeric_limits<double>::epsilon())` [protected],[inherited]

Definition at line 229 of file estimator.h.

Referenced by mappel::IterativeMaximizer< Model >::convergence_test(), mappel::IterativeMaximizer< Model >::get_stats(), mappel::NewtonDiagonalMaximizer< 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 = sqrt(std::numeric_limits<double>::epsilon())` `[protected]`, `[inherited]`

Definition at line 228 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`.

8.35.6.4 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::exit_counts` `[protected]`, `[inherited]`

Definition at line 240 of file estimator.h.

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

8.35.6.5 `template<class Model > double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` `[protected]`, `[inherited]`

Definition at line 231 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 242 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 233 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 225 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `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 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model>::anneal()`, `mappel::IterativeMaximizer< Model>::backtrack()`, `mappel::ThreadedEstimator< Model>::clear_stats()`, `mappel::CGaussMLE< Model>::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate()`, `mappel::IterativeMaximizer< Model>::compute_estimate()`, `mappel::CGaussMLE< Model>::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, `mappel::IterativeMaximizer< Model>::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model>::maximize()`, `mappel::NewtonMaximizer< Model>::maximize()`, `mappel::QuasiNewtonMaximizer< Model>::maximize()`, and `mappel::TrustRegionMaximizer< Model>::maximize()`.

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 128 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::IterativeMaximizer< Model>::NumExitCodes = 7` [static],
[inherited]

Definition at line 197 of file estimator.h.

8.35.6.15 `template<class Model> int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected],
[inherited]

Definition at line 237 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 239 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 238 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 236 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 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

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

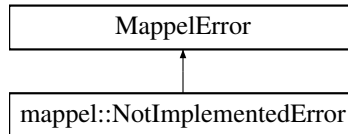
- [estimator.h](#)
- [estimator_impl.h](#)

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:

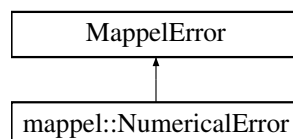
- [util.h](#)

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_::OMPEXceptionCatcher< IntType > Class Template Reference

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

Public Types

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

Public Member Functions

- [OMPEXceptionCatcher](#) ()
- [OMPEXceptionCatcher](#) ([Strategy](#) strategy_)
- void [rethrow](#) () const
- template<class Function , class... Parameters>
void [run](#) (Function func, Parameters...params)

Static Public Member Functions

- static void [setGlobalDefaultStrategy](#) ([Strategy](#) s)

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::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::methods←::openmp::estimate_mcmc_posterior_stack()`, `mappel::methods::openmp::estimate_mcmc_sample_stack()`, `mappel←::ThreadedEstimator< Model >::estimate_profile_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::OMPEXceptionCatcher< IntType >::run ( Function func, Parameters... params )
[inline]
```

Definition at line 57 of file OMPEXceptionCatcher.h.

References `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::Abort`, `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::Continue`, `omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::DoNotTry`, and `omp_exception_catcher::impl::OMPEXceptionCatcher< 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::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::methods::openmp::estimate_mcmc_posterior_stack()`, `mappel::methods::openmp::estimate_mcmc_sample_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_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::OMPEXceptionCatcher<
IntType >::setGlobalDefaultStrategy ( Strategy s ) [inline],[static]
```

Definition at line 49 of file OMPEXceptionCatcher.h.

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

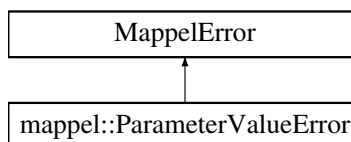
- [OMPEXceptionCatcher.h](#)

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:

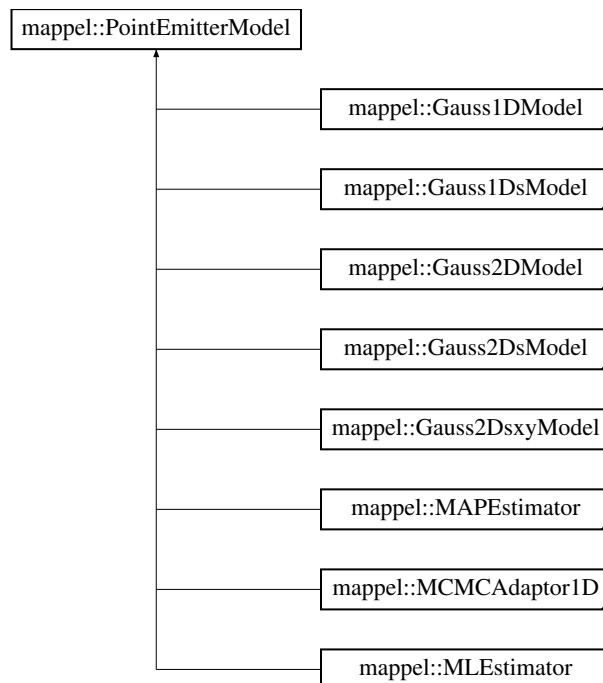
- [util.h](#)

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` ↵ 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 priorsInitialized with a prior as a `PriorHessian::`↵ CompositeDist object, this sets the dimensionality (`num_params`) and `num_hyperparams`, and the associated descriptions.

Box-type bounding constraints are controlled with the [set_bounds\(\)](#) method.

Of note some of the common MCMC variables are rooted here in the inheritance tree.

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

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 30 of file `PointEmitterModel.cpp`.

8.40.3.2 mappel::PointEmitterModel::PointEmitterModel (const CompositeDist & prior_) [explicit], [protected]

Definition at line 42 of file `PointEmitterModel.cpp`.

8.40.3.3 `mappel::PointEmitterModel::PointEmitterModel (CompositeDist && prior_) [explicit],[protected]`

Definition at line 36 of file PointEmitterModel.cpp.

8.40.3.4 `mappel::PointEmitterModel::PointEmitterModel (const PointEmitterModel & o) [protected]`

Definition at line 48 of file PointEmitterModel.cpp.

References `prior`.

8.40.3.5 `mappel::PointEmitterModel::PointEmitterModel (PointEmitterModel && o) [protected]`

Definition at line 54 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 247 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 264 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 306 of file PointEmitterModel.cpp.

References `bounded_theta()`, and `check_param_shape()`.

8.40.4.4 `void mappel::PointEmitterModel::check_param_shape (const ParamT & theta) const`

Definition at line 166 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 175 of file PointEmitterModel.cpp.

References `num_params`.

8.40.4.6 `void mappel::PointEmitterModel::check_psf_sigma (double psf_sigma) const`

Definition at line 184 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 196 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 119 of file PointEmitterModel.cpp.

References `mappel::rng_manager`.

8.40.4.19 ParallelRngManagerT & mappel::PointEmitterModel::get_rng_manager () [static]

Definition at line 114 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.40.4.20 StatsT mappel::PointEmitterModel::get_stats () const

Definition at line 124 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.

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 97 of file PointEmitterModel.cpp.

References default_max_I.

Referenced by mappel::Gauss1DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss1DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss1DsModel::make_default_prior_normal_position(), mappel::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 91 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 84 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 103 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 60 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 67 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 275 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 316 of file `PointEmitterModel.cpp`.

References `check_param_shape()`, 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 212 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 225 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 157 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 148 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 109 of file PointEmitterModel.cpp.

References mappel::rng_manager.

8.40.4.51 void mappel::PointEmitterModel::set_ubound (const ParamT & ubound)

Definition at line 236 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 256 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 296 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_I = INFINITY` [static]

Default emitter intensity mean

Definition at line 59 of file PointEmitterModel.h.

Referenced by `make_prior_component_intensity()`.

8.40.5.6 `const double mappel::PointEmitterModel::default_mean_I = 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:

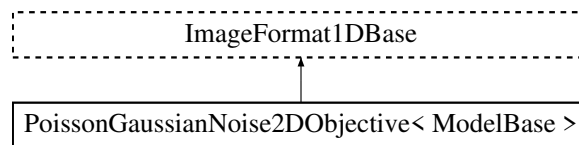
- [PointEmitterModel.h](#)
- [PointEmitterModel.cpp](#)

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:

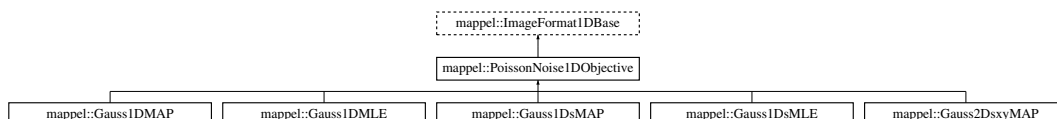
- [PoissonGaussianNoise2DObjective.h](#)
- [PoissonGaussianNoise2DObjective.cpp](#)

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:

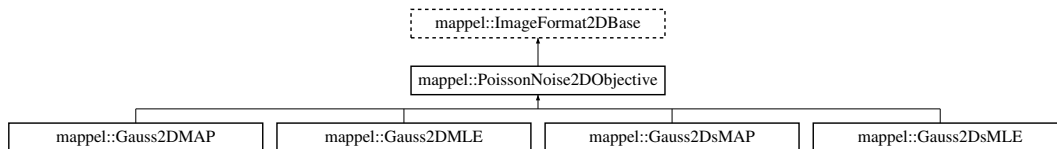
- [PoissonNoise1DObjective.h](#)
- [PoissonNoise1DObjective.cpp](#)

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::Gauss2DsMA←
P()`, `mappel::Gauss2DsMLE::Gauss2DsMLE()`, `mappel::Gauss2DsModel::Gauss2DsModel()`, `mappel::Image←
Format2DBase::get_num_pixels()`, `mappel::ImageFormat2DBase::get_size()`, `mappel::ImageFormat2DBase::get←
_stats()`, `mappel::methods::likelihood::grad()`, `mappel::methods::likelihood::grad2()`, `mappel::methods::likelihood←
::debug::grad_components()`, `mappel::methods::likelihood::hessian()`, `mappel::methods::likelihood::debug::hessian←
_components()`, `mappel::Gauss2DModel::initial_theta_estimate()`, `mappel::Gauss2DsModel::initial_theta_estimate()`,

mappel::methods::likelihood::llh(), mappel::methods::likelihood::debug::llh_components(), mappel::Gauss2DModel::make_default_prior_beta_position(), mappel::Gauss2DsModel::make_default_prior_beta_position(), mappel::Gauss2DModel::make_default_prior_normal_position(), mappel::Gauss2DsModel::make_default_prior_normal_position(), mappel::ImageFormat2DBase::make_image(), mappel::ImageFormat2DBase::make_image_stack(), mappel::Gauss2DModel::make_internal_1Dsum_estimator(), mappel::Gauss2DsModel::make_internal_1Dsum_estimator(), mappel::Gauss2DModel::make_prior_beta_position(), mappel::Gauss2DsModel::make_prior_beta_position(), mappel::Gauss2DModel::make_prior_normal_position(), mappel::Gauss2DsModel::make_prior_normal_position(), mappel::methods::model_image(), mappel::ImageFormat2DBase::operator=(), mappel::methods::likelihood::rllh(), mappel::methods::likelihood::debug::rllh_components(), mappel::ImageFormat2DBase::set_size(), mappel::Gauss2DModel::set_size(), mappel::Gauss2DsModel::set_size(), mappel::methods::simulate_image(), mappel::methods::simulate_image_from_model(), mappel::Gauss2DModel::Stencil::Stencil(), mappel::Gauss2DsModel::Stencil::Stencil(), mappel::Gauss2DModel::update_internal_1Dsum_estimators(), and mappel::Gauss2DsModel::update_internal_1Dsum_estimators().

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

- [PoissonNoise2DObjective.h](#)
- [PoissonNoise2DObjective.cpp](#)

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 $\text{kappa_bg} * (\log(\text{kappa_bg}) - 1 / \text{mean_bg} - \log(\text{mean_bg})) - \text{lgamma}(\text{kappa_bg})$

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 $\text{kappa_l} * (\log(\text{kappa_l}) - 1 / \text{mean_l} - \log(\text{mean_l})) - \text{lgamma}(\text{kappa_l})$

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 $-2 \cdot \lgamma(\beta_x) - \lgamma(2 \cdot \beta_x)$

Definition at line 28 of file PriorMAP1DObjective.h.

Referenced by prior_log_likelihood().

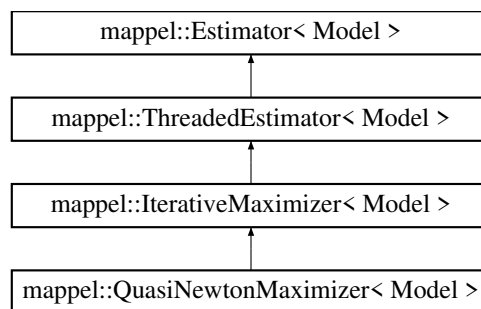
The documentation for this class was generated from the following file:

- [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](#) = 99, [ExitCode::MaxIter](#) = 6, [ExitCode::MaxBacktracks](#) = 5, [ExitCode::TrustRegionRadius](#) = 4,
[ExitCode::GradRatio](#) = 3, [ExitCode::FunctionChange](#) = 2, [ExitCode::StepSize](#) = 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` = 7

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_walltime` (ClockT::time_point start_walltime, int nimages)

Protected Attributes

- int [max_iterations](#)
- double [epsilon](#) = sqrt(std::numeric_limits<double>::epsilon())
- double [delta](#) = sqrt(std::numeric_limits<double>::epsilon())
- 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](#) [exit_counts](#)
- [IdxVecT](#) [last_backtrack_idx](#)
- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.

8.45.1 Detailed Description

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

Definition at line 350 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 352 of file estimator.h.

8.45.3 Member Enumeration Documentation

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

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Error

Definition at line 198 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 354 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 495 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::alpha`, `mappel::IterativeMaximizer< Model >::convergence_`, `test()`, `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_parameter_scalar`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::im`, `mappel::IterativeMaximizer< Model >::max_backtracks`, `mappel::IterativeMaximizer< Model >::MaxBacktracks`, `mappel::Estimator< Model >::model`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_backtrack()`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `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 469 of file estimator_impl.h.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::exit_counts`, `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 142 of file estimator_impl.h.

References `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 593 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::Error](#), [mappel::IterativeMaximizer< Model >::MaximizerData::get_theta_sequence\(\)](#), [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::print_image\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::record_exit\(\)](#), [mappel::IterativeMaximizer< Model >::record_run_statistics\(\)](#), [mappel::IterativeMaximizer< Model >::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model >::MaximizerData::stencil\(\)](#), and [mappel::IterativeMaximizer< Model >::MaximizerData::theta\(\)](#).

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 618 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::Error](#), [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::IterativeMaximizer< Model >::MaximizerData::record_exit\(\)](#), [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 637 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::Error](#), [mappel::IterativeMaximizer< Model >::maximize\(\)](#), [mappel::Estimator< Model >::model](#), [mappel::IterativeMaximizer< Model >::MaximizerData::record_exit\(\)](#), [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 571 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::IterativeMaximizer< Model >::MaximizerData::fixed_parameter_scalar`, `mappel::IterativeMaximizer< Model >::FunctionChange`, `mappel::IterativeMaximizer< Model >::MaximizerData::grad`, `mappel::IterativeMaximizer< Model >::MaximizerData::has_fixed_parameters`, `mappel::IterativeMaximizer< Model >::MaximizerData::record_exit()`, `mappel::IterativeMaximizer< Model >::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model >::MaximizerData::saved_theta()`, `mappel::IterativeMaximizer< Model >::StepSize`, and `mappel::IterativeMaximizer< Model >::MaximizerData::theta()`.

Referenced by `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, and `mappel::QuasiNewtonMaximizer< Model >::maximize()`.

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

Definition at line 59 of file estimator_impl.h.

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

Referenced by `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 77 of file estimator_impl.h.

References `mappel::methods::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 68 of file estimator_impl.h.

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

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 85 of file estimator_impl.h.

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 97 of file estimator_impl.h.

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

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 106 of file estimator_impl.h.

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 117 of file estimator_impl.h.

References `mappel::methods::observed_information()`, 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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

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 222 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

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 246 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

8.45.5.18 `template<class Model> StatsT mappel::IterativeMaximizer< Model>::get_debug_stats () [virtual],
[inherited]`

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

Definition at line 453 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 45 of file `estimator_impl.h`.

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 424 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model>::delta`, `mappel::IterativeMaximizer< Model>::epsilon`, `mappel::IterativeMaximizer< Model>::Error`, `mappel::IterativeMaximizer< Model>::exit_counts`, `mappel::IterativeMaximizer< Model>::FunctionChange`, `mappel::ThreadedEstimator< Model>::get_stats()`, `mappel::IterativeMaximizer< Model>::GradRatio`, `mappel::IterativeMaximizer< Model>::max_backtracks`, `mappel::IterativeMaximizer< Model>::max_←_iterations`, `mappel::IterativeMaximizer< Model>::MaxBacktracks`, `mappel::IterativeMaximizer< Model>::MaxIter`, `mappel::ThreadedEstimator< Model>::mtx`, `mappel::Estimator< Model>::num_estimations`, `mappel::Iterative←Maximizer< Model>::StepSize`, `mappel::IterativeMaximizer< Model>::total_backtracks`, `mappel::IterativeMaximizer< Model>::total_der_evals`, `mappel::IterativeMaximizer< Model>::total_fun_evals`, `mappel::IterativeMaximizer< Model>::total_iterations`, and `mappel::IterativeMaximizer< Model>::TrustRegionRadius`.

Referenced by `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 655 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 752 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::methods::objective::grad()`, `mappel::methods::objective::hessian()`, `mappel::is_positive_definite()`, `mappel::IterativeMaximizer< Model >::max_iterations`, and `mappel::Estimator< Model >::model`.

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 357 of file `estimator.h`.

8.45.5.29 `template<class Model > void mappel::IterativeMaximizer< Model >::record_run_statistics (const MaximizerData & data)`
`[protected], [inherited]`

Definition at line 482 of file `estimator_impl.h`.

References `mappel::IterativeMaximizer< Model >::MaximizerData::exit_code`, `mappel::IterativeMaximizer< Model >::exit_counts`, `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()`.

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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 232 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 = sqrt(std::numeric_limits<double>::epsilon())` [protected], [inherited]

Definition at line 229 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< 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 = sqrt(std::numeric_limits<double>::epsilon())` [protected], [inherited]

Definition at line 228 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`.

8.45.6.4 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::exit_counts` [protected], [inherited]

Definition at line 240 of file estimator.h.

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

8.45.6.5 `template<class Model> double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` [protected],
[inherited]

Definition at line 231 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` [protected],
[inherited]

Definition at line 242 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 233 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 225 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `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 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::maximize()`.

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 128 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::IterativeMaximizer< Model >::NumExitCodes = 7` [static],
[inherited]

Definition at line 197 of file estimator.h.

8.45.6.15 `template<class Model > int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected],
[inherited]

Definition at line 237 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 239 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 238 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 236 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 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

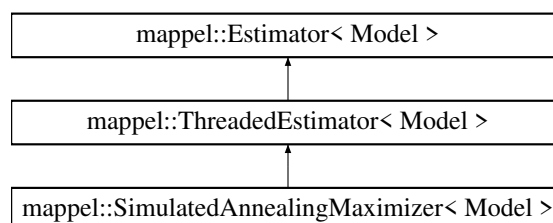
The documentation for this class was generated from the following files:

- [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 Member Functions

- `std::string name () const`
- `SimulatedAnnealingMaximizer (Model &model)`
- `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, ParamVecT< Model > &theta, 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)`
- `StatsT get_stats ()`
- `StatsT get_debug_stats ()`
- `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 max_iterations =500`

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)`
- `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.

8.46.1 Detailed Description

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

Definition at line 176 of file estimator.h.

8.46.2 Constructor & Destructor Documentation

8.46.2.1 `template<class Model > mappel::SimulatedAnnealingMaximizer< Model >::SimulatedAnnealingMaximizer (Model & model) [inline]`

Definition at line 183 of file estimator.h.

References `mappel::Estimator< Model >::compute_estimate()`, `mappel::Estimator< Model >::compute_estimate_↵ debug()`, and `mappel::methods::objective::rllh()`.

8.46.3 Member Function Documentation

8.46.3.1 `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 1319 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::IterativeMaximizer< Model >::max_↵ iterations`, `mappel::Estimator< Model >::model`, and `mappel::methods::objective::rllh()`.

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

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

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

Definition at line 287 of file estimator_impl.h.

References `mappel::cgauss_heuristic_compute_estimate()`, `mappel::Estimator< Model >::clear_stats()`, `mappel::↵ Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, and `mappel::methods::objective↵ ::rllh()`.

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

8.46.3.3 `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 142 of file estimator_impl.h.

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

8.46.3.4 `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 1300 of file estimator_impl.h.

References `mappel::Estimator< Model>::model`, and `mappel::IterativeMaximizer< Model>::MaximizerData::rllh`.

8.46.3.5 `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 1309 of file estimator_impl.h.

References `mappel::Estimator< Model>::model`, and `mappel::IterativeMaximizer< Model>::MaximizerData::rllh`.

8.46.3.6 `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 151 of file estimator_impl.h.

References `mappel::Estimator< Model>::name()`.

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

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

Definition at line 59 of file estimator_impl.h.

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

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

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

Definition at line 77 of file estimator_impl.h.

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

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

Definition at line 68 of file estimator_impl.h.

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

8.46.3.10 `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 85 of file estimator_impl.h.

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

Definition at line 97 of file estimator_impl.h.

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

8.46.3.12 `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 106 of file estimator_impl.h.

8.46.3.13 `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 117 of file estimator_impl.h.

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

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

8.46.3.14 `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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

8.46.3.15 `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 222 of file estimator_impl.h.

References [mappel::Estimator< Model>::model](#), [mappel::ThreadedEstimator< Model>::num_threads](#), [mappel::Estimator< Model>::record_walltime\(\)](#), [omp_exception_catcher::impl::OMPExceptionCatcher< IntType>::rethrow\(\)](#), and [omp_exception_catcher::impl::OMPExceptionCatcher< IntType>::run\(\)](#).

8.46.3.16 `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 246 of file estimator_impl.h.

References [mappel::Estimator< Model>::model](#), [mappel::ThreadedEstimator< Model>::num_threads](#), [mappel::Estimator< Model>::record_walltime\(\)](#), [omp_exception_catcher::impl::OMPExceptionCatcher< IntType>::rethrow\(\)](#), and [omp_exception_catcher::impl::OMPExceptionCatcher< IntType>::run\(\)](#).

8.46.3.17 `template<class Model> StatsT mappel::ThreadedEstimator< Model>::get_debug_stats () [virtual], [inherited]`

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

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

Definition at line 281 of file estimator_impl.h.

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

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

Definition at line 45 of file estimator_impl.h.

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

8.46.3.19 `template<class Model> StatsT mappel::ThreadedEstimator< Model>::get_stats () [virtual], [inherited]`

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

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

Definition at line 271 of file estimator_impl.h.

References [mappel::Estimator< Model>::get_stats\(\)](#), [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::CGaussMLE< Model>::get_stats\(\)](#), and [mappel::IterativeMaximizer< Model>::get_stats\(\)](#).

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

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

Definition at line 182 of file estimator.h.

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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

8.46.4 Member Data Documentation

8.46.4.1 `template<class Model > double mappel::SimulatedAnnealingMaximizer< Model >::cooling_rate=1.02`

Definition at line 179 of file estimator.h.

8.46.4.2 `template<class Model > int mappel::SimulatedAnnealingMaximizer< Model >::max_iterations=500`

Definition at line 180 of file estimator.h.

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

Definition at line 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by [mappel::SimulatedAnnealingMaximizer< Model >::anneal\(\)](#), [mappel::IterativeMaximizer< Model >::backtrack\(\)](#), [mappel::ThreadedEstimator< Model >::clear_stats\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate\(\)](#), [mappel::CGaussMLE< Model >::compute_estimate_debug\(\)](#), [mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_estimate_debug\(\)](#), [mappel::IterativeMaximizer< Model >::compute_profile_estimate\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_max_stack\(\)](#), [mappel::ThreadedEstimator< Model >::estimate_profile_stack\(\)](#), [mappel::IterativeMaximizer< Model >::local_maximize\(\)](#), [mappel::NewtonDiagonalMaximizer< Model >::maximize\(\)](#), [mappel::NewtonMaximizer< Model >::maximize\(\)](#), [mappel::QuasiNewtonMaximizer< Model >::maximize\(\)](#), and [mappel::TrustRegionMaximizer< Model >::maximize\(\)](#).

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 128 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()`.

8.46.4.8 `template<class Model > double mappel::SimulatedAnnealingMaximizer< Model >::T_init = 100.`

Definition at line 178 of file estimator.h.

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

Definition at line 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

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::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::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_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::Gauss2DsxxyModel::Stencil::x () const [inline]`

Definition at line 50 of file Gauss2DsxxyModel.h.

References `theta`.

8.49.4.7 `double mappel::Gauss2DsxxyModel::Stencil::y () const [inline]`

Definition at line 51 of file Gauss2DsxxyModel.h.

References `theta`.

8.49.5 Friends And Related Function Documentation

8.49.5.1 `std::ostream& operator<< (std::ostream & out, const Gauss2DsxxyModel::Stencil & s) [friend]`

Referenced by `sigmaY()`.

8.49.6 Member Data Documentation

8.49.6.1 `bool mappel::Gauss2DsxxyModel::Stencil::derivatives_computed =false`

Definition at line 34 of file Gauss2DsxxyModel.h.

8.49.6.2 `VecT mappel::Gauss2DsxxyModel::Stencil::dx`

Definition at line 39 of file Gauss2DsxxyModel.h.

8.49.6.3 `VecT mappel::Gauss2DsxxyModel::Stencil::DX`

Definition at line 42 of file Gauss2DsxxyModel.h.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

8.49.6.4 `VecT mappel::Gauss2DsxxyModel::Stencil::DXS`

Definition at line 44 of file Gauss2DsxxyModel.h.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad()`, `mappel::Gauss2DsxxyModel::pixel_grad2()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

8.49.6.5 `VecT mappel::Gauss2DsxxyModel::Stencil::DXS2`

Definition at line 45 of file Gauss2DsxxyModel.h.

Referenced by `mappel::Gauss2DsxxyModel::pixel_grad2()`, and `mappel::Gauss2DsxxyModel::pixel_hess()`.

8.49.6.6 VecT mappel::Gauss2DsxyModel::Stencil::DXSX

Definition at line 43 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_hess().

8.49.6.7 VecT mappel::Gauss2DsxyModel::Stencil::dy

Definition at line 39 of file Gauss2DsxyModel.h.

8.49.6.8 VecT mappel::Gauss2DsxyModel::Stencil::DY

Definition at line 42 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_grad(), and mappel::Gauss2DsxyModel::pixel_hess().

8.49.6.9 VecT mappel::Gauss2DsxyModel::Stencil::DYS

Definition at line 44 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_grad(), mappel::Gauss2DsxyModel::pixel_grad2(), and mappel::Gauss2DsxyModel::pixel_hess().

8.49.6.10 VecT mappel::Gauss2DsxyModel::Stencil::DYS2

Definition at line 45 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_grad2(), and mappel::Gauss2DsxyModel::pixel_hess().

8.49.6.11 VecT mappel::Gauss2DsxyModel::Stencil::DYSX

Definition at line 43 of file Gauss2DsxyModel.h.

8.49.6.12 VecT mappel::Gauss2DsxyModel::Stencil::DYSY

Definition at line 46 of file Gauss2DsxyModel.h.

Referenced by mappel::Gauss2DsxyModel::pixel_hess().

8.49.6.13 VecT mappel::Gauss2DsxyModel::Stencil::Gx

Definition at line 40 of file Gauss2DsxyModel.h.

8.49.6.14 VecT mappel::Gauss2DsxyModel::Stencil::Gy

Definition at line 40 of file Gauss2DsxyModel.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 (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().

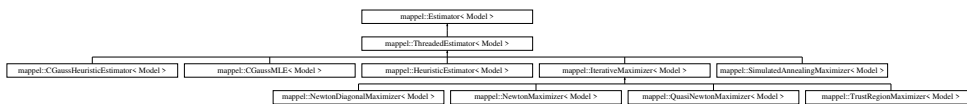
The documentation for this class was generated from the following files:

- [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 Member Functions

- [ThreadedEstimator](#) (Model &model)
- void [estimate_max_stack](#) (const [ModelDataStackT](#)< Model > &im, const [ParamVecT](#)< Model > &theta_init, [ParamVecT](#)< Model > &theta, [VecT](#) &rllh, [CubeT](#) &obsI)
- 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)

Protected Member Functions

- 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 seperately 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.

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 differnt method for estimate_stack pure virtual member function. For now all other (CPU) estimators inherit from [ThreadedEstimator](#).

Definition at line 113 of file estimator.h.

8.52.2 Constructor & Destructor Documentation

8.52.2.1 template<class Model > mappel::ThreadedEstimator< Model >::ThreadedEstimator (Model & model)

Definition at line 213 of file estimator_impl.h.

8.52.3 Member Function Documentation

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

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

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

Definition at line 287 of file `estimator_impl.h`.

References `mappel::cgauss_heuristic_compute_estimate()`, `mappel::Estimator< Model >::clear_stats()`, `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, and `mappel::methods::objective::rllh()`.

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

8.52.3.2 `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::HeuristicEstimator< Model >::name()`, `mappel::CGaussHeuristicEstimator< Model >::name()`, `mappel::CGaussMLE< Model >::name()`, `mappel::SimulatedAnnealingMaximizer< Model >::SimulatedAnnealingMaximizer()`, and `mappel::Estimator< Model >::~~Estimator()`.

8.52.3.3 `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 142 of file `estimator_impl.h`.

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

8.52.3.4 `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) [inline], [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 192 of file `estimator_impl.h`.

Referenced by `mappel::CGaussMLE< Model >::name()`, `mappel::SimulatedAnnealingMaximizer< Model >::SimulatedAnnealingMaximizer()`, and `mappel::Estimator< Model >::~~Estimator()`.

8.52.3.5 `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 151 of file estimator_impl.h.

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

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

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

Definition at line 59 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

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

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

Definition at line 77 of file estimator_impl.h.

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

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

Definition at line 68 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

8.52.3.9 `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 85 of file estimator_impl.h.

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

Definition at line 97 of file estimator_impl.h.

References [mappel::methods::estimate_max\(\)](#).

8.52.3.11 `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 106 of file estimator_impl.h.

8.52.3.12 `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 117 of file estimator_impl.h.

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

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

8.52.3.13 `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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

8.52.3.14 `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 222 of file estimator_impl.h.

References `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::Estimator< Model>::record_walltime()`, `omp_exception_catcher::impl::OMPExcceptionCatcher< IntType>::rethrow()`, and `omp_exception_catcher::impl::OMPExcceptionCatcher< IntType>::run()`.

8.52.3.15 `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 246 of file estimator_impl.h.

References `mappel::Estimator< Model>::model`, `mappel::ThreadedEstimator< Model>::num_threads`, `mappel::Estimator< Model>::record_walltime()`, `omp_exception_catcher::impl::OMPExcceptionCatcher< IntType>::rethrow()`, and `omp_exception_catcher::impl::OMPExcceptionCatcher< IntType>::run()`.

8.52.3.16 `template<class Model> StatsT mappel::ThreadedEstimator< Model>::get_debug_stats () [virtual]`

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

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

Definition at line 281 of file estimator_impl.h.

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

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

Definition at line 45 of file estimator_impl.h.

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

8.52.3.18 `template<class Model > StatsT mappel::ThreadedEstimator< Model >::get_stats ()` [virtual]

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

Reimplemented in `mappel::IterativeMaximizer< Model >`, and `mappel::CGaussMLE< Model >`.

Definition at line 271 of file estimator_impl.h.

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

8.52.3.19 `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 >::compute_profile_estimate()`, and `mappel::Estimator< Model >::~~Estimator()`.

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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

8.52.4 Member Data Documentation

8.52.4.1 `template<class Model > int mappel::ThreadedEstimator< Model >::max_threads` [protected]

Definition at line 127 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model>::anneal()`, `mappel::IterativeMaximizer< Model>::backtrack()`, `mappel::ThreadedEstimator< Model>::clear_stats()`, `mappel::CGaussMLE< Model>::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate()`, `mappel::IterativeMaximizer< Model>::compute_estimate()`, `mappel::CGaussMLE< Model>::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model>::compute_profile_estimate()`, `mappel::ThreadedEstimator< Model>::estimate_max_stack()`, `mappel::ThreadedEstimator< Model>::estimate_profile_stack()`, `mappel::IterativeMaximizer< Model>::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model>::maximize()`, `mappel::NewtonMaximizer< Model>::maximize()`, `mappel::QuasiNewtonMaximizer< Model>::maximize()`, and `mappel::TrustRegionMaximizer< Model>::maximize()`.

8.52.4.3 `template<class Model> std::mutex mappel::ThreadedEstimator< Model>::mtx` [protected]

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

8.52.4.5 `template<class Model> int mappel::ThreadedEstimator< Model>::num_threads` [protected]

Definition at line 128 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()`.

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

Definition at line 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model>::get_stats()`.

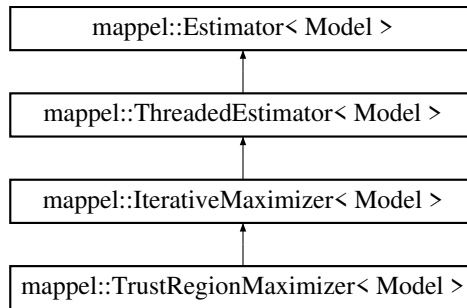
The documentation for this class was generated from the following files:

- [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](#) = 99, [ExitCode::MaxIter](#) = 6, [ExitCode::MaxBacktracks](#) = 5, [ExitCode::TrustRegionRadius](#) = 4,
[ExitCode::GradRatio](#) = 3, [ExitCode::FunctionChange](#) = 2, [ExitCode::StepSize](#) = 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_← 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](#) &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](#) = 7

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_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](#) = sqrt(std::numeric_limits<double>::epsilon())
- double [delta](#) = sqrt(std::numeric_limits<double>::epsilon())
- 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 exit_counts](#)
- [IdxVecT last_backtrack_idx](#)
- int [max_threads](#)
- int [num_threads](#)
- std::mutex [mtx](#)
- Model & [model](#)
- int [num_estimations](#) = 0
- double [total_walltime](#) = 0.

8.53.1 Detailed Description

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

Definition at line 364 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 366 of file estimator.h.

8.53.3 Member Enumeration Documentation

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

Enumerator

Unassigned
MaxIter
MaxBacktracks
TrustRegionRadius
GradRatio
FunctionChange
StepSize
Error

Definition at line 198 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 381 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 495 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model>::alpha`, `mappel::IterativeMaximizer< Model>::convergence_`, `test()`, `mappel::IterativeMaximizer< Model>::MaximizerData::fixed_parameter_scalar`, `mappel::IterativeMaximizer< Model>::MaximizerData::grad`, `mappel::IterativeMaximizer< Model>::MaximizerData::has_fixed_parameters`, `mappel::IterativeMaximizer< Model>::MaximizerData::im`, `mappel::IterativeMaximizer< Model>::max_backtracks`, `mappel::IterativeMaximizer< Model>::MaxBacktracks`, `mappel::Estimator< Model>::model`, `mappel::IterativeMaximizer< Model>::MaximizerData::record_backtrack()`, `mappel::IterativeMaximizer< Model>::MaximizerData::record_exit()`, `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 1069 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 469 of file estimator_impl.h.

References `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::IterativeMaximizer< Model >::exit_counts`, `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 1033 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 1110 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. $\text{sign}(\text{grad}^2)$ is not important

Definition at line 994 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 142 of file estimator_impl.h.

References `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 593 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model>::Error`, `mappel::IterativeMaximizer< Model>::MaximizerData::get_theta_sequence()`, `mappel::IterativeMaximizer< Model>::maximize()`, `mappel::Estimator< Model>::model`, `mappel::print_image()`, `mappel::IterativeMaximizer< Model>::MaximizerData::record_exit()`, `mappel::IterativeMaximizer< Model>::record_run_statistics()`, `mappel::IterativeMaximizer< Model>::MaximizerData::rllh`, `mappel::IterativeMaximizer< Model>::MaximizerData::stencil()`, and `mappel::IterativeMaximizer< Model>::MaximizerData::theta()`.

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 618 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model>::Error`, `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::IterativeMaximizer< Model>::MaximizerData::record_exit()`, `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 1007 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 637 of file `estimator_impl.h`.

References [mappel::IterativeMaximizer< Model>::Error](#), [mappel::IterativeMaximizer< Model>::maximize\(\)](#), [mappel::Estimator< Model>::model](#), [mappel::IterativeMaximizer< Model>::MaximizerData::record_exit\(\)](#), [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 571 of file `estimator_impl.h`.

References [mappel::IterativeMaximizer< Model>::delta](#), [mappel::IterativeMaximizer< Model>::epsilon](#), [mappel::IterativeMaximizer< Model>::MaximizerData::fixed_parameter_scalar](#), [mappel::IterativeMaximizer< Model>::FunctionChange](#), [mappel::IterativeMaximizer< Model>::MaximizerData::grad](#), [mappel::IterativeMaximizer< Model>::MaximizerData::has_fixed_parameters](#), [mappel::IterativeMaximizer< Model>::MaximizerData::record_exit\(\)](#), [mappel::IterativeMaximizer< Model>::MaximizerData::rllh](#), [mappel::IterativeMaximizer< Model>::MaximizerData::saved_theta\(\)](#), [mappel::IterativeMaximizer< Model>::StepSize](#), and [mappel::IterativeMaximizer< Model>::MaximizerData::theta\(\)](#).

Referenced by [mappel::IterativeMaximizer< Model>::backtrack\(\)](#), [mappel::NewtonDiagonalMaximizer< Model>::maximize\(\)](#), [mappel::NewtonMaximizer< Model>::maximize\(\)](#), and [mappel::QuasiNewtonMaximizer< Model>::maximize\(\)](#).

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

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

References [mappel::methods::estimate_max\(\)](#).

Referenced by [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 77 of file `estimator_impl.h`.

References [mappel::methods::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 68 of file `estimator_impl.h`.

References [mappel::methods::estimate_max\(\)](#).

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 85 of file estimator_impl.h.

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 97 of file estimator_impl.h.

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

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 106 of file estimator_impl.h.

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 117 of file estimator_impl.h.

References `mappel::methods::observed_information()`, 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 129 of file estimator_impl.h.

References `mappel::methods::openmp::estimate_max_stack()`.

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 222 of file estimator_impl.h.

References `mappel::Estimator< Model >::model`, `mappel::ThreadedEstimator< Model >::num_threads`, `mappel::Estimator< Model >::record_walltime()`, `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::rethrow()`, and `omp_exception_catcher::impl::OMPExceptionCatcher< IntType >::run()`.

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 246 of file estimator_impl.h.

References [mappel::Estimator< Model >::model](#), [mappel::ThreadedEstimator< Model >::num_threads](#), [mappel::Estimator< Model >::record_walltime\(\)](#), [omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::rethrow\(\)](#), and [omp_exception_catcher::impl::OMPEXceptionCatcher< IntType >::run\(\)](#).

8.53.5.23 `template<class Model > StatsT mappel::IterativeMaximizer< Model >::get_debug_stats () [virtual], [inherited]`

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

Definition at line 453 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 45 of file estimator_impl.h.

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 424 of file estimator_impl.h.

References [mappel::IterativeMaximizer< Model >::delta](#), [mappel::IterativeMaximizer< Model >::epsilon](#), [mappel::IterativeMaximizer< Model >::Error](#), [mappel::IterativeMaximizer< Model >::exit_counts](#), [mappel::IterativeMaximizer< Model >::FunctionChange](#), [mappel::ThreadedEstimator< Model >::get_stats\(\)](#), [mappel::IterativeMaximizer< Model >::GradRatio](#), [mappel::IterativeMaximizer< Model >::max_backtracks](#), [mappel::IterativeMaximizer< Model >::max_iterations](#), [mappel::IterativeMaximizer< Model >::MaxBacktracks](#), [mappel::IterativeMaximizer< Model >::MaxIter](#), [mappel::ThreadedEstimator< Model >::mtx](#), [mappel::Estimator< Model >::num_estimations](#), [mappel::IterativeMaximizer< Model >::StepSize](#), [mappel::IterativeMaximizer< Model >::total_backtracks](#), [mappel::IterativeMaximizer< Model >::total_der_evals](#), [mappel::IterativeMaximizer< Model >::total_fun_evals](#), [mappel::IterativeMaximizer< Model >::total_iterations](#), and [mappel::IterativeMaximizer< Model >::TrustRegionRadius](#).

Referenced by [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 655 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 827 of file estimator_impl.h.

References `mappel::IterativeMaximizer< Model >::delta`, `mappel::IterativeMaximizer< Model >::epsilon`, `mappel::methods::objective::hessian()`, `mappel::Estimator< Model >::model`, 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 384 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 1019 of file estimator_impl.h.

8.53.5.35 `template<class Model> void mappel::IterativeMaximizer< Model >::record_run_statistics (const MaximizerData & data)` [protected], [inherited]

Definition at line 482 of file estimator_impl.h.

References mappel::IterativeMaximizer< Model >::MaximizerData::exit_code, mappel::IterativeMaximizer< Model >::exit_counts, 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().

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

Definition at line 203 of file estimator_impl.h.

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

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

Definition at line 49 of file estimator_impl.h.

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

8.53.5.38 `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 1219 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.

8.53.5.39 `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 1127 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 232 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 379 of file estimator.h.

8.53.6.3 `template<class Model > double mappel::IterativeMaximizer< Model >::delta = sqrt(std::numeric_limits<double>::epsilon())` `[protected]`, `[inherited]`

Definition at line 229 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< 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 372 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 371 of file estimator.h.

8.53.6.6 `template<class Model > const double mappel::TrustRegionMaximizer< Model >::delta_increase = 2` `[static]`

Definition at line 373 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 378 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 377 of file estimator.h.

8.53.6.9 `template<class Model > double mappel::IterativeMaximizer< Model >::epsilon =
sqrt(std::numeric_limits<double>::epsilon())` `[protected]`, `[inherited]`

Definition at line 228 of file estimator.h.

Referenced by `mappel::IterativeMaximizer< Model >::convergence_test()`, `mappel::IterativeMaximizer< Model >::get_stats()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::TrustRegionMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::solve_restricted_step_length_newton()`.

8.53.6.10 `template<class Model > IdxVecT mappel::IterativeMaximizer< Model >::exit_counts` `[protected]`, `[inherited]`

Definition at line 240 of file estimator.h.

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

8.53.6.11 `template<class Model > double mappel::IterativeMaximizer< Model >::lambda_min = 0.05` `[protected]`, `[inherited]`

Definition at line 231 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`s `[protected]`, `[inherited]`

Definition at line 242 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 233 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 225 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `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 376 of file estimator.h.

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

Definition at line 127 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 375 of file estimator.h.

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

Definition at line 97 of file estimator.h.

Referenced by `mappel::SimulatedAnnealingMaximizer< Model >::anneal()`, `mappel::IterativeMaximizer< Model >::backtrack()`, `mappel::ThreadedEstimator< Model >::clear_stats()`, `mappel::CGaussMLE< Model >::compute_estimate()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate()`, `mappel::IterativeMaximizer< Model >::compute_estimate()`, `mappel::CGaussMLE< Model >::compute_estimate_debug()`, `mappel::SimulatedAnnealingMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_estimate_debug()`, `mappel::IterativeMaximizer< Model >::compute_profile_estimate()`, `mappel::ThreadedEstimator< Model >::estimate_max_stack()`, `mappel::ThreadedEstimator< Model >::estimate_profile_stack()`, `mappel::IterativeMaximizer< Model >::local_maximize()`, `mappel::NewtonDiagonalMaximizer< Model >::maximize()`, `mappel::NewtonMaximizer< Model >::maximize()`, `mappel::QuasiNewtonMaximizer< Model >::maximize()`, and `mappel::TrustRegionMaximizer< Model >::maximize()`.

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

Definition at line 129 of file estimator.h.

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

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

Definition at line 100 of file estimator.h.

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

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

Definition at line 128 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::IterativeMaximizer< Model >::NumExitCodes = 7` [static],
[inherited]

Definition at line 197 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 368 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 369 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 370 of file estimator.h.

8.53.6.26 `template<class Model> int mappel::IterativeMaximizer< Model >::total_backtracks = 0` [protected],
[inherited]

Definition at line 237 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 239 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 238 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 236 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 101 of file estimator.h.

Referenced by `mappel::ThreadedEstimator< Model >::get_stats()`.

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)

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)

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::Gauss1DMAP](#)
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 Gauss1DMAP.

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/PoissonNoiselDObjective.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/MCMCA adaptor1Ds.h"
```

Classes

- class `mappel::Gauss1DsModel`
Base class for 1D Gaussian PSF with variable Gaussian sigma (standard deviation) measured in units of pixels.
- class `mappel::Gauss1DsModel::Stencil`
Stencil for 1D variable-sigma models.

Namespaces

- `mappel`

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/MCMCAdaptor2D.h"
#include "Mappel/Gauss1DMAP.h"
```

Classes

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

Namespaces

- [mappel](#)

Functions

- `template<class Model >`
`std::enable_if< std::is_base_of< Gauss2DModel, Model >::value, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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, StencilT< 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 Gauss2DsxMAP.h File Reference

The class declaration and inline and templated functions for Gauss2DsxMAP.

```
#include "Mappel/PoissonNoise2DObjective.h"
#include "Mappel/MAPEstimator.h"
#include "Mappel/Gauss1DModel.h"
#include "Mappel/model_methods.h"
```

Classes

- class [mappel::Gauss2DsxMAP](#)
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 Gauss2DsxMAP.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017

9.31 Gauss2DsxModel.h File Reference

The class declaration and inline and templated functions for Gauss2DsxModel.

```
#include "Mappel/PointEmitterModel.h"
#include "Mappel/ImageFormat2DBase.h"
#include "Mappel/Gauss1DsMAP.h"
```

Classes

- class [mappel::Gauss2DsxModel](#)
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::Gauss2DsxModel::Stencil](#)
Stencil for 2D free-sigma (astigmatic) models.

Namespaces

- [mappel](#)

Functions

- `template<class Model >`
`std::enable_if< std::is_base_of< Gauss2DsxModel, Model >::value, StencilT< 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, StencilT< 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, StencilT< 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 OMPExcptionCatcher.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_::OMPExcptionCatcher< IntType >](#)

Namespaces

- [omp_exception_catcher](#)
- [omp_exception_catcher::impl_](#)

Typedefs

- using [omp_exception_catcher::OMPEXceptionCatcher](#) = impl_::OMPEXceptionCatcher< 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 OMPEXceptionCatcher::Strategies enum.

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

Example useage: OMPEXceptionCatcher catcher(OMPEXceptionCatcher<>::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, type-
name 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 type-
name 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, [101](#)
- Abort
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [678](#)
- alpha
 - mappel::IterativeMaximizer, [534](#)
 - mappel::NewtonDiagonalMaximizer, [660](#)
 - mappel::NewtonMaximizer, [672](#)
 - mappel::QuasiNewtonMaximizer, [724](#)
 - mappel::TrustRegionMaximizer, [776](#)
- anneal
 - mappel::SimulatedAnnealingMaximizer, [729](#)
- aposteriori_objective
 - mappel::methods, [53](#)
- ArrayShapeError
 - mappel::ArrayShapeError, [86](#)
- ArraySizeError
 - mappel::ArraySizeError, [87](#)
- backtrack
 - mappel::IterativeMaximizer, [528](#)
 - mappel::NewtonDiagonalMaximizer, [654](#)
 - mappel::NewtonMaximizer, [666](#)
 - mappel::QuasiNewtonMaximizer, [718](#)
 - mappel::TrustRegionMaximizer, [768](#)
- backtrack_idxes
 - mappel::IterativeMaximizer::MaximizerData, [558](#)
- beta2_prior_grad
 - mappel, [28](#)
- beta2_prior_grad2
 - mappel, [28](#)
- beta_prior_grad
 - mappel, [28](#)
- beta_prior_grad2
 - mappel, [28](#)
- bg
 - mappel::Gauss1DModel::Stencil, [736](#)
 - mappel::Gauss1DsModel::Stencil, [740](#)
 - mappel::Gauss2DModel::Stencil, [755](#)
 - mappel::Gauss2DsModel::Stencil, [749](#)
 - mappel::Gauss2DsxyModel::Stencil, [744](#)
- BoolVecT
 - mappel, [25](#)
- BoolT
 - mappel, [25](#)
- bound_step
 - mappel::TrustRegionMaximizer, [768](#)
- bound_theta
 - mappel::Gauss1DMAP, [113](#)
 - mappel::Gauss1DMLE, [141](#)
- mappel::Gauss1DModel, [168](#)
- mappel::Gauss1DsMAP, [196](#)
- mappel::Gauss1DsMLE, [224](#)
- mappel::Gauss1DsModel, [252](#)
- mappel::Gauss2DMAP, [280](#)
- mappel::Gauss2DMLE, [310](#)
- mappel::Gauss2DModel, [339](#)
- mappel::Gauss2DsMAP, [368](#)
- mappel::Gauss2DsMLE, [400](#)
- mappel::Gauss2DsModel, [432](#)
- mappel::Gauss2DsxyMAP, [462](#)
- mappel::Gauss2DsxyModel, [489](#)
- mappel::MAPEstimator, [541](#)
- mappel::MCMCAdaptor1Ds, [582](#)
- mappel::MCMCAdaptor1D, [564](#)
- mappel::MCMCAdaptor2Ds, [618](#)
- mappel::MCMCAdaptor2D, [600](#)
- mappel::MLEstimator, [638](#)
- mappel::PointEmitterModel, [684](#)
- boundary_stepback_min_kappa
 - mappel::TrustRegionMaximizer, [776](#)
- bounded_theta
 - mappel::Gauss1DMAP, [113](#)
 - mappel::Gauss1DMLE, [141](#)
 - mappel::Gauss1DModel, [168](#)
 - mappel::Gauss1DsMAP, [196](#)
 - mappel::Gauss1DsMLE, [224](#)
 - mappel::Gauss1DsModel, [252](#)
 - mappel::Gauss2DMAP, [280](#)
 - mappel::Gauss2DMLE, [310](#)
 - mappel::Gauss2DModel, [339](#)
 - mappel::Gauss2DsMAP, [368](#)
 - mappel::Gauss2DsMLE, [400](#)
 - mappel::Gauss2DsModel, [432](#)
 - mappel::Gauss2DsxyMAP, [462](#)
 - mappel::Gauss2DsxyModel, [489](#)
 - mappel::MAPEstimator, [541](#)
 - mappel::MCMCAdaptor1Ds, [582](#)
 - mappel::MCMCAdaptor1D, [564](#)
 - mappel::MCMCAdaptor2Ds, [618](#)
 - mappel::MCMCAdaptor2D, [600](#)
 - mappel::MLEstimator, [638](#)
 - mappel::PointEmitterModel, [684](#)
- bounded_theta_stack
 - mappel::Gauss1DMAP, [113](#)
 - mappel::Gauss1DMLE, [141](#)
 - mappel::Gauss1DModel, [169](#)
 - mappel::Gauss1DsMAP, [196](#)
 - mappel::Gauss1DsMLE, [224](#)
 - mappel::Gauss1DsModel, [252](#)
 - mappel::Gauss2DMAP, [280](#)

- mappel::Gauss2DMLE, 310
- mappel::Gauss2DModel, 339
- mappel::Gauss2DsMAP, 369
- mappel::Gauss2DsMLE, 401
- mappel::Gauss2DsModel, 432
- mappel::Gauss2DsxyMAP, 463
- mappel::Gauss2DsxyModel, 489
- mappel::MAPEstimator, 541
- mappel::MCMCAdaptor1Ds, 582
- mappel::MCMCAdaptor1D, 564
- mappel::MCMCAdaptor2Ds, 618
- mappel::MCMCAdaptor2D, 600
- mappel::MLEstimator, 639
- mappel::PointEmitterModel, 684
- bounds_epsilon
 - mappel::Gauss1DMAP, 128
 - mappel::Gauss1DMLE, 156
 - mappel::Gauss1DModel, 183
 - mappel::Gauss1DsMAP, 211
 - mappel::Gauss1DsMLE, 239
 - mappel::Gauss1DsModel, 267
 - mappel::Gauss2DMAP, 295
 - mappel::Gauss2DMLE, 325
 - mappel::Gauss2DModel, 354
 - mappel::Gauss2DsMAP, 386
 - mappel::Gauss2DsMLE, 418
 - mappel::Gauss2DsModel, 449
 - mappel::Gauss2DsxyMAP, 477
 - mappel::Gauss2DsxyModel, 501
 - mappel::MAPEstimator, 550
 - mappel::MCMCAdaptor1Ds, 591
 - mappel::MCMCAdaptor1D, 574
 - mappel::MCMCAdaptor2Ds, 628
 - mappel::MCMCAdaptor2D, 609
 - mappel::MLEstimator, 647
 - mappel::PointEmitterModel, 692
- CGaussHeuristicEstimator
 - mappel::CGaussHeuristicEstimator, 88
- CGaussMLE
 - mappel::CGaussMLE, 95
- cgauss_compute_estimate
 - mappel, 28
- cgauss_compute_estimate_debug
 - mappel, 28, 29
- cgauss_heuristic_compute_estimate
 - mappel, 29
- check_image_shape
 - mappel::Gauss1DMAP, 113
 - mappel::Gauss1DMLE, 141
 - mappel::Gauss1DModel, 169
 - mappel::Gauss1DsMAP, 196
 - mappel::Gauss1DsMLE, 225
 - mappel::Gauss1DsModel, 252
- mappel::Gauss2DMAP, 280
- mappel::Gauss2DMLE, 310
- mappel::Gauss2DModel, 339
- mappel::Gauss2DsMAP, 369
- mappel::Gauss2DsMLE, 401
- mappel::Gauss2DsModel, 433
- mappel::Gauss2DsxyMAP, 463
- mappel::Gauss2DsxyModel, 489
- mappel::ImageFormat1DBase, 515
- mappel::ImageFormat2DBase, 522
- mappel::PoissonNoise1DObjective, 701
- mappel::PoissonNoise2DObjective, 708
- check_lower_bound_hyperparameter
 - mappel, 29
- check_param_shape
 - mappel::Gauss1DMAP, 114
 - mappel::Gauss1DMLE, 142
 - mappel::Gauss1DModel, 169
 - mappel::Gauss1DsMAP, 197
 - mappel::Gauss1DsMLE, 225
 - mappel::Gauss1DsModel, 252
 - mappel::Gauss2DMAP, 280, 281
 - mappel::Gauss2DMLE, 310, 311
 - mappel::Gauss2DModel, 340
 - mappel::Gauss2DsMAP, 369
 - mappel::Gauss2DsMLE, 401
 - mappel::Gauss2DsModel, 433
 - mappel::Gauss2DsxyMAP, 463
 - mappel::Gauss2DsxyModel, 489, 490
 - mappel::MAPEstimator, 541, 542
 - mappel::MCMCAdaptor1Ds, 582
 - mappel::MCMCAdaptor1D, 565
 - mappel::MCMCAdaptor2Ds, 618
 - mappel::MCMCAdaptor2D, 600
 - mappel::MLEstimator, 639
 - mappel::PointEmitterModel, 684
- check_positive_hyperparameter
 - mappel, 29
- check_psf_sigma
 - mappel::Gauss1DMAP, 114
 - mappel::Gauss1DMLE, 142
 - mappel::Gauss1DModel, 169, 170
 - mappel::Gauss1DsMAP, 197
 - mappel::Gauss1DsMLE, 225
 - mappel::Gauss1DsModel, 253
 - mappel::Gauss2DMAP, 281
 - mappel::Gauss2DMLE, 311
 - mappel::Gauss2DModel, 340
 - mappel::Gauss2DsMAP, 369, 370
 - mappel::Gauss2DsMLE, 401, 402
 - mappel::Gauss2DsModel, 433
 - mappel::Gauss2DsxyMAP, 463, 464
 - mappel::Gauss2DsxyModel, 490
 - mappel::MAPEstimator, 542

- mappel::MCMCAdaptor1Ds, 583
- mappel::MCMCAdaptor1D, 565
- mappel::MCMCAdaptor2Ds, 619
- mappel::MCMCAdaptor2D, 601
- mappel::MLEstimator, 639
- mappel::PointEmitterModel, 685
- check_size
 - mappel::Gauss1DModel, 114
 - mappel::Gauss1DModel, 142
 - mappel::Gauss1DModel, 170
 - mappel::Gauss1DsMAP, 197
 - mappel::Gauss1DsMLE, 226
 - mappel::Gauss1DsModel, 253
 - mappel::Gauss2DModel, 281
 - mappel::Gauss2DModel, 311
 - mappel::Gauss2DModel, 340
 - mappel::Gauss2DsMAP, 370
 - mappel::Gauss2DsMLE, 402
 - mappel::Gauss2DsModel, 434
 - mappel::Gauss2DsxyMAP, 464
 - mappel::Gauss2DsxyModel, 490
 - mappel::ImageFormat1DBase, 515
 - mappel::ImageFormat2DBase, 522
 - mappel::PoissonNoise1DObjective, 701
 - mappel::PoissonNoise2DObjective, 709
- 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, 88
 - mappel::CGaussMLE, 95
 - mappel::Estimator, 102
 - mappel::HeuristicEstimator, 507
 - mappel::IterativeMaximizer, 528
 - mappel::NewtonDiagonalMaximizer, 654
 - mappel::NewtonMaximizer, 666
 - mappel::QuasiNewtonMaximizer, 718
 - mappel::SimulatedAnnealingMaximizer, 729
 - mappel::ThreadedEstimator, 760
 - mappel::TrustRegionMaximizer, 769
- ClockT
 - estimator.h, 783
- compute_D_scale
 - mappel::TrustRegionMaximizer, 769
- compute_bound_scaling_vec
 - mappel::TrustRegionMaximizer, 769
- compute_cauchy_point
 - mappel::TrustRegionMaximizer, 769
- compute_derivatives
 - mappel::Gauss1DModel::Stencil, 736
 - mappel::Gauss1DsModel::Stencil, 740
 - mappel::Gauss2DModel::Stencil, 755
 - mappel::Gauss2DsModel::Stencil, 749
 - mappel::Gauss2DsxyModel::Stencil, 744
- compute_estimate
 - mappel::CGaussHeuristicEstimator, 88
 - mappel::CGaussMLE, 95
 - mappel::Estimator, 102
 - mappel::HeuristicEstimator, 507
 - mappel::IterativeMaximizer, 529
 - mappel::NewtonDiagonalMaximizer, 654
 - mappel::NewtonMaximizer, 667
 - mappel::QuasiNewtonMaximizer, 718
 - mappel::SimulatedAnnealingMaximizer, 729, 730
 - mappel::ThreadedEstimator, 760
 - mappel::TrustRegionMaximizer, 769, 770
- compute_estimate_debug
 - mappel::CGaussHeuristicEstimator, 89
 - mappel::CGaussMLE, 95
 - mappel::Estimator, 102
 - mappel::HeuristicEstimator, 507
 - mappel::IterativeMaximizer, 529
 - mappel::NewtonDiagonalMaximizer, 655
 - mappel::NewtonMaximizer, 667
 - mappel::QuasiNewtonMaximizer, 719
 - mappel::SimulatedAnnealingMaximizer, 730
 - mappel::ThreadedEstimator, 760
 - mappel::TrustRegionMaximizer, 770
- compute_initial_trust_radius
 - mappel::TrustRegionMaximizer, 770
- compute_max_sigma_ratio
 - mappel::Gauss2DsMAP, 370
 - mappel::Gauss2DsMLE, 402
 - mappel::Gauss2DsModel, 434
 - mappel::Gauss2DsxyModel, 490
- compute_profile_estimate
 - mappel::CGaussHeuristicEstimator, 89
 - mappel::CGaussMLE, 96
 - mappel::Estimator, 102
 - mappel::HeuristicEstimator, 508
 - mappel::IterativeMaximizer, 529
 - mappel::NewtonDiagonalMaximizer, 655
 - mappel::NewtonMaximizer, 667
 - mappel::QuasiNewtonMaximizer, 719
 - mappel::SimulatedAnnealingMaximizer, 730
 - mappel::ThreadedEstimator, 760
 - mappel::TrustRegionMaximizer, 770

- Continue
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [678](#)
- convergence_test
 - mappel::IterativeMaximizer, [530](#)
 - mappel::NewtonDiagonalMaximizer, [655](#)
 - mappel::NewtonMaximizer, [668](#)
 - mappel::QuasiNewtonMaximizer, [719](#)
 - mappel::TrustRegionMaximizer, [771](#)
- cooling_rate
 - mappel::SimulatedAnnealingMaximizer, [733](#)
- CoordIdxT
 - PoissonGaussianNoise2DObjective, [696](#)
- CoordStackT
 - PoissonGaussianNoise2DObjective, [696](#)
- CoordT
 - PoissonGaussianNoise2DObjective, [696](#)
- 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, [558](#)
- DXS2
 - mappel::Gauss1DsModel::Stencil, [741](#)
 - mappel::Gauss2DsModel::Stencil, [751](#)
 - mappel::Gauss2DsxyModel::Stencil, [745](#)
- DXSX
 - mappel::Gauss1DsModel::Stencil, [741](#)
 - mappel::Gauss2DsModel::Stencil, [751](#)
 - mappel::Gauss2DsxyModel::Stencil, [745](#)
- DXS
 - mappel::Gauss1DModel::Stencil, [737](#)
 - mappel::Gauss1DsModel::Stencil, [741](#)
 - mappel::Gauss2DModel::Stencil, [756](#)
 - mappel::Gauss2DsModel::Stencil, [751](#)
 - mappel::Gauss2DsxyModel::Stencil, [745](#)
- DYS2
 - mappel::Gauss2DsModel::Stencil, [752](#)
 - mappel::Gauss2DsxyModel::Stencil, [746](#)
- DYSX
 - mappel::Gauss2DsxyModel::Stencil, [746](#)
- DYSY
 - mappel::Gauss2DsModel::Stencil, [752](#)
- mappel::Gauss2DsxyModel::Stencil, [746](#)
- DYS
 - mappel::Gauss2DModel::Stencil, [757](#)
 - mappel::Gauss2DsModel::Stencil, [751](#)
 - mappel::Gauss2DsxyModel::Stencil, [746](#)
- debug_internal_sum_model_x
 - mappel::Gauss2DMAP, [281](#)
 - mappel::Gauss2DMLE, [311](#)
 - mappel::Gauss2DModel, [340](#)
 - mappel::Gauss2DsMAP, [370](#)
 - mappel::Gauss2DsMLE, [402](#)
 - mappel::Gauss2DsModel, [434](#)
- debug_internal_sum_model_y
 - mappel::Gauss2DMAP, [281](#)
 - mappel::Gauss2DMLE, [311](#)
 - mappel::Gauss2DModel, [341](#)
 - mappel::Gauss2DsMAP, [370](#)
 - mappel::Gauss2DsMLE, [402](#)
 - mappel::Gauss2DsModel, [434](#)
- default_alpha_sigma
 - mappel::Gauss1DMAP, [128](#)
 - mappel::Gauss1DMLE, [156](#)
 - mappel::Gauss1DModel, [183](#)
 - mappel::Gauss1DsMAP, [211](#)
 - mappel::Gauss1DsMLE, [239](#)
 - mappel::Gauss1DsModel, [267](#)
 - mappel::Gauss2DMAP, [295](#)
 - mappel::Gauss2DMLE, [325](#)
 - mappel::Gauss2DModel, [354](#)
 - mappel::Gauss2DsMAP, [386](#)
 - mappel::Gauss2DsMLE, [418](#)
 - mappel::Gauss2DsModel, [449](#)
 - mappel::Gauss2DsxyMAP, [477](#)
 - mappel::Gauss2DsxyModel, [501](#)
 - mappel::MAPEstimator, [550](#)
 - mappel::MCMCAdaptor1Ds, [591](#)
 - mappel::MCMCAdaptor1D, [574](#)
 - mappel::MCMCAdaptor2Ds, [628](#)
 - mappel::MCMCAdaptor2D, [609](#)
 - mappel::MLEstimator, [647](#)
 - mappel::PointEmitterModel, [692](#)
- default_beta_pos
 - mappel::Gauss1DMAP, [128](#)
 - mappel::Gauss1DMLE, [156](#)
 - mappel::Gauss1DModel, [183](#)
 - mappel::Gauss1DsMAP, [211](#)
 - mappel::Gauss1DsMLE, [240](#)
 - mappel::Gauss1DsModel, [267](#)
 - mappel::Gauss2DMAP, [295](#)
 - mappel::Gauss2DMLE, [325](#)
 - mappel::Gauss2DModel, [354](#)
 - mappel::Gauss2DsMAP, [386](#)
 - mappel::Gauss2DsMLE, [418](#)
 - mappel::Gauss2DsModel, [449](#)

- mappel::Gauss2DsxyMAP, [477](#)
- mappel::Gauss2DsxyModel, [501](#)
- mappel::MAPEstimator, [550](#)
- mappel::MCMCAdaptor1Ds, [592](#)
- mappel::MCMCAdaptor1D, [574](#)
- mappel::MCMCAdaptor2Ds, [628](#)
- mappel::MCMCAdaptor2D, [610](#)
- mappel::MLEstimator, [647](#)
- mappel::PointEmitterModel, [692](#)
- default_intensity_kappa
 - mappel::Gauss1DMAP, [128](#)
 - mappel::Gauss1DMLE, [156](#)
 - mappel::Gauss1DModel, [183](#)
 - mappel::Gauss1DsMAP, [212](#)
 - mappel::Gauss1DsMLE, [240](#)
 - mappel::Gauss1DsModel, [267](#)
 - mappel::Gauss2DMAP, [296](#)
 - mappel::Gauss2DMLE, [326](#)
 - mappel::Gauss2DModel, [354](#)
 - mappel::Gauss2DsMAP, [386](#)
 - mappel::Gauss2DsMLE, [418](#)
 - mappel::Gauss2DsModel, [449](#)
 - mappel::Gauss2DsxyMAP, [477](#)
 - mappel::Gauss2DsxyModel, [501](#)
 - mappel::MAPEstimator, [550](#)
 - mappel::MCMCAdaptor1Ds, [592](#)
 - mappel::MCMCAdaptor1D, [574](#)
 - mappel::MCMCAdaptor2Ds, [628](#)
 - mappel::MCMCAdaptor2D, [610](#)
 - mappel::MLEstimator, [647](#)
 - mappel::PointEmitterModel, [692](#)
- default_max_l
 - mappel::Gauss1DMAP, [128](#)
 - mappel::Gauss1DMLE, [156](#)
 - mappel::Gauss1DModel, [183](#)
 - 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, [477](#)
 - mappel::Gauss2DsxyModel, [501](#)
 - mappel::MAPEstimator, [550](#)
 - mappel::MCMCAdaptor1Ds, [592](#)
 - mappel::MCMCAdaptor1D, [574](#)
 - mappel::MCMCAdaptor2Ds, [628](#)
 - mappel::MCMCAdaptor2D, [610](#)
 - mappel::MLEstimator, [647](#)
 - mappel::PointEmitterModel, [693](#)
- default_mean_l
 - 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, [387](#)
- default_pixel_mean_bg
 - 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, [502](#)
 - mappel::MAPEstimator, [550](#)
 - mappel::MCMCAdaptor1Ds, [592](#)
 - mappel::MCMCAdaptor1D, [575](#)
 - mappel::MCMCAdaptor2Ds, [628](#)
 - mappel::MCMCAdaptor2D, [610](#)
 - mappel::MLEstimator, [647](#)
 - mappel::PointEmitterModel, [693](#)
- default_sigma_pos
 - mappel::Gauss1DMAP, [129](#)
 - mappel::Gauss1DMLE, [157](#)
 - mappel::Gauss1DModel, [184](#)
 - mappel::Gauss1DsMAP, [212](#)
 - mappel::Gauss1DsMLE, [240](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DMAP, [296](#)
 - mappel::Gauss2DMLE, [326](#)
 - mappel::Gauss2DModel, [355](#)
 - mappel::Gauss2DsMAP, [387](#)

- mappel::Gauss2DsMLE, [419](#)
- mappel::Gauss2DsModel, [450](#)
- mappel::Gauss2DsxyMAP, [478](#)
- mappel::Gauss2DsxyModel, [502](#)
- mappel::MAPEstimator, [551](#)
- mappel::MCMCAdaptor1Ds, [592](#)
- mappel::MCMCAdaptor1D, [575](#)
- mappel::MCMCAdaptor2Ds, [629](#)
- mappel::MCMCAdaptor2D, [610](#)
- mappel::MLEstimator, [648](#)
- mappel::PointEmitterModel, [693](#)
- DefaultPriorType
 - mappel::Gauss1DModel, [129](#)
 - mappel::Gauss1DModel, [157](#)
 - mappel::Gauss1DModel, [184](#)
 - mappel::Gauss1DsMAP, [212](#)
 - mappel::Gauss1DsMLE, [241](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DModel, [296](#)
 - mappel::Gauss2DModel, [326](#)
 - mappel::Gauss2DModel, [355](#)
 - mappel::Gauss2DsMAP, [387](#)
 - mappel::Gauss2DsMLE, [419](#)
 - mappel::Gauss2DsModel, [450](#)
 - mappel::Gauss2DsxyMAP, [478](#)
- DefaultSeperableInitEstimator
 - mappel::Gauss1DModel, [129](#)
 - mappel::Gauss1DModel, [157](#)
 - mappel::Gauss1DModel, [184](#)
 - mappel::Gauss1DsMAP, [213](#)
 - mappel::Gauss1DsMLE, [241](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DModel, [297](#)
 - mappel::Gauss2DModel, [327](#)
 - mappel::Gauss2DModel, [355](#)
 - mappel::Gauss2DsMAP, [387](#)
 - mappel::Gauss2DsMLE, [419](#)
 - mappel::Gauss2DsModel, [450](#)
 - mappel::Gauss2DsxyMAP, [478](#)
 - mappel::Gauss2DsxyModel, [502](#)
 - mappel::MAPEstimator, [551](#)
 - mappel::MCMCAdaptor1Ds, [593](#)
 - mappel::MCMCAdaptor1D, [575](#)
 - mappel::MCMCAdaptor2Ds, [629](#)
 - mappel::MCMCAdaptor2D, [611](#)
 - mappel::MLEstimator, [648](#)
 - mappel::PointEmitterModel, [693](#)
- delta
 - mappel::IterativeMaximizer, [534](#)
 - mappel::NewtonDiagonalMaximizer, [660](#)
 - mappel::NewtonMaximizer, [672](#)
 - mappel::QuasiNewtonMaximizer, [724](#)
 - mappel::TrustRegionMaximizer, [776](#)
- delta_decrease
 - mappel::TrustRegionMaximizer, [776](#)
- delta_decrease_min
 - mappel::TrustRegionMaximizer, [776](#)
- delta_increase
 - mappel::TrustRegionMaximizer, [776](#)
- delta_init_max
 - mappel::TrustRegionMaximizer, [776](#)
- delta_init_min
 - mappel::TrustRegionMaximizer, [776](#)
- derivatives_computed
 - mappel::Gauss1DModel::Stencil, [737](#)
 - mappel::Gauss1DsModel::Stencil, [741](#)
 - mappel::Gauss2DModel::Stencil, [756](#)
 - mappel::Gauss2DsModel::Stencil, [750](#)
 - mappel::Gauss2DsxyModel::Stencil, [745](#)
- display.cpp, [780](#)
- display.h, [781](#)
- DoNotTry
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [678](#)
- DX
 - mappel::Gauss1DModel::Stencil, [737](#)
 - mappel::Gauss1DsModel::Stencil, [741](#)
 - mappel::Gauss2DModel::Stencil, [756](#)
 - mappel::Gauss2DsModel::Stencil, [751](#)
 - mappel::Gauss2DsxyModel::Stencil, [745](#)
- dx
 - mappel::Gauss1DModel::Stencil, [737](#)
 - mappel::Gauss1DsModel::Stencil, [741](#)
 - mappel::Gauss2DModel::Stencil, [756](#)
 - mappel::Gauss2DsModel::Stencil, [750](#)
 - mappel::Gauss2DsxyModel::Stencil, [745](#)
- DY
 - mappel::Gauss2DModel::Stencil, [757](#)
 - mappel::Gauss2DsModel::Stencil, [751](#)
 - mappel::Gauss2DsxyModel::Stencil, [746](#)
- dy
 - mappel::Gauss2DModel::Stencil, [756](#)
 - mappel::Gauss2DsModel::Stencil, [751](#)
 - mappel::Gauss2DsxyModel::Stencil, [746](#)
- enable_all_cpus
 - mappel, [31](#)
- EnableIfSubclassT
 - mappel, [25](#)
- epsilon
 - mappel::IterativeMaximizer, [534](#)
 - mappel::NewtonDiagonalMaximizer, [660](#)
 - mappel::NewtonMaximizer, [672](#)
 - mappel::QuasiNewtonMaximizer, [724](#)
 - mappel::TrustRegionMaximizer, [776](#)
- Error
 - mappel::IterativeMaximizer, [528](#)
 - mappel::NewtonDiagonalMaximizer, [653](#)

- mappel::NewtonMaximizer, 666
- mappel::QuasiNewtonMaximizer, 717
- mappel::TrustRegionMaximizer, 768
- 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, 89, 90
 - mappel::CGaussMLE, 96, 97
 - mappel::Estimator, 103
 - mappel::HeuristicEstimator, 508, 509
 - mappel::IterativeMaximizer, 530, 531
 - mappel::NewtonDiagonalMaximizer, 656
 - mappel::NewtonMaximizer, 668, 669
 - mappel::QuasiNewtonMaximizer, 720
 - mappel::SimulatedAnnealingMaximizer, 730, 731
 - mappel::ThreadedEstimator, 761
 - mappel::TrustRegionMaximizer, 771, 772
 - mappel::methods, 54, 55
- estimate_max_debug
 - mappel::CGaussHeuristicEstimator, 90
 - mappel::CGaussMLE, 97
 - mappel::Estimator, 103
 - mappel::HeuristicEstimator, 509
 - mappel::IterativeMaximizer, 531
 - mappel::NewtonDiagonalMaximizer, 657
 - mappel::NewtonMaximizer, 669
 - mappel::QuasiNewtonMaximizer, 721
 - mappel::SimulatedAnnealingMaximizer, 731
 - mappel::ThreadedEstimator, 761
 - mappel::TrustRegionMaximizer, 772
 - mappel::methods::debug, 62
- estimate_max_stack
 - mappel::CGaussHeuristicEstimator, 90
 - mappel::CGaussMLE, 97
 - mappel::Estimator, 104
 - mappel::HeuristicEstimator, 509
- mappel::IterativeMaximizer, 531
- mappel::NewtonDiagonalMaximizer, 657
- mappel::NewtonMaximizer, 669
- mappel::QuasiNewtonMaximizer, 721
- mappel::SimulatedAnnealingMaximizer, 731
- mappel::ThreadedEstimator, 762
- mappel::TrustRegionMaximizer, 772
- mappel::methods::openmp, 81
- estimate_mcmc_posterior
 - mappel::methods, 55, 56
- 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, 90
 - mappel::CGaussMLE, 97
 - mappel::Estimator, 104
 - mappel::HeuristicEstimator, 509
 - mappel::IterativeMaximizer, 531
 - mappel::NewtonDiagonalMaximizer, 657
 - mappel::NewtonMaximizer, 669
 - mappel::QuasiNewtonMaximizer, 721
 - mappel::SimulatedAnnealingMaximizer, 732
 - mappel::ThreadedEstimator, 762
 - mappel::TrustRegionMaximizer, 772
- estimate_sample_posterior
 - mappel::mcmc, 49
- Estimator
 - mappel::Estimator, 101
- estimator.h, 782
 - ClockT, 783
- estimator_impl.h, 783
- estimator_names
 - mappel::Gauss1DMAP, 129
 - mappel::Gauss1DMLE, 157
 - mappel::Gauss1DsMAP, 213
 - mappel::Gauss1DsMLE, 241
 - mappel::Gauss2DMAP, 297
 - mappel::Gauss2DMLE, 327
 - mappel::Gauss2DsMAP, 387
 - mappel::Gauss2DsMLE, 419
 - mappel::Gauss2DsxyMAP, 478
 - mappel::PoissonNoise1DObjective, 703
 - mappel::PoissonNoise2DObjective, 711
 - PoissonGaussianNoise2DObjective, 697
- estimator_statics.cpp, 784
- eta_bg

- mappel::Gauss1DMap, [130](#)
- mappel::Gauss1DMLE, [158](#)
- mappel::Gauss1DModel, [184](#)
- 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::MCMCAdaptor1Ds, [593](#)
- mappel::MCMCAdaptor1D, [575](#)
- mappel::MCMCAdaptor2Ds, [629](#)
- mappel::MCMCAdaptor2D, [611](#)
- eta_l
 - 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::MCMCAdaptor1Ds, [593](#)
 - mappel::MCMCAdaptor1D, [575](#)
 - mappel::MCMCAdaptor2Ds, [629](#)
 - mappel::MCMCAdaptor2D, [611](#)
- eta_sigma
 - mappel::Gauss1DsMap, [213](#)
 - mappel::Gauss1DsMLE, [241](#)
 - mappel::Gauss1DsModel, [268](#)
 - mappel::Gauss2DsMap, [388](#)
 - mappel::Gauss2DsMLE, [420](#)
 - mappel::Gauss2DsModel, [451](#)
 - mappel::MCMCAdaptor1Ds, [593](#)
 - mappel::MCMCAdaptor2Ds, [629](#)
- eta_x
 - mappel::Gauss1DMap, [130](#)
 - mappel::Gauss1DMLE, [158](#)
 - mappel::Gauss1DModel, [185](#)
 - mappel::Gauss1DsMap, [213](#)
 - mappel::Gauss1DsMLE, [242](#)
 - mappel::Gauss1DsModel, [269](#)
 - mappel::Gauss2DMap, [297](#)
 - mappel::Gauss2DMLE, [327](#)
 - mappel::Gauss2DModel, [356](#)
 - mappel::Gauss2DsMap, [388](#)
 - mappel::Gauss2DsMLE, [420](#)
 - mappel::Gauss2DsModel, [451](#)
 - mappel::Gauss2DsxyMap, [479](#)
 - mappel::MCMCAdaptor1Ds, [593](#)
 - mappel::MCMCAdaptor1D, [576](#)
 - mappel::MCMCAdaptor2Ds, [630](#)
 - mappel::MCMCAdaptor2D, [611](#)
- eta_y
 - mappel::Gauss2DMap, [297](#)
 - mappel::Gauss2DMLE, [327](#)
 - mappel::Gauss2DModel, [356](#)
 - mappel::Gauss2DsMap, [388](#)
 - mappel::Gauss2DsMLE, [420](#)
 - mappel::Gauss2DsModel, [451](#)
 - mappel::MCMCAdaptor2Ds, [630](#)
 - mappel::MCMCAdaptor2D, [611](#)
- exit_code
 - mappel::IterativeMaximizer::MaximizerData, [558](#)
- exit_counts
 - mappel::IterativeMaximizer, [534](#)
 - mappel::NewtonDiagonalMaximizer, [660](#)
 - mappel::NewtonMaximizer, [673](#)
 - mappel::QuasiNewtonMaximizer, [724](#)
 - mappel::TrustRegionMaximizer, [777](#)
- ExitCode
 - mappel::IterativeMaximizer, [528](#)
 - mappel::NewtonDiagonalMaximizer, [653](#)
 - mappel::NewtonMaximizer, [666](#)
 - mappel::QuasiNewtonMaximizer, [717](#)
 - mappel::TrustRegionMaximizer, [768](#)
- 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, [829](#)

- fisher_information_stack
 - mappel, 33
- fixed_parameter_scalar
 - mappel::IterativeMaximizer::MaximizerData, 558
- FunctionChange
 - mappel::IterativeMaximizer, 528
 - mappel::NewtonDiagonalMaximizer, 653
 - mappel::NewtonMaximizer, 666
 - mappel::QuasiNewtonMaximizer, 717
 - mappel::TrustRegionMaximizer, 768
- gamma_prior_grad
 - mappel, 33
- gamma_prior_grad2
 - mappel, 33
- Gauss1DModel.cpp, 784
- Gauss1DModel.h, 785
- Gauss1DModel
 - mappel::Gauss1DModel, 112, 113
- Gauss1DModel.cpp, 785
- Gauss1DModel.h, 786
- Gauss1DModel
 - mappel::Gauss1DModel, 140, 141
- Gauss1DModel
 - mappel::Gauss1DModel, 168
- Gauss1DModel.cpp, 787
- Gauss1DModel.h, 787
- Gauss1DSumModelT
 - mappel::Gauss2DModel, 277
 - mappel::Gauss2DModel, 307
 - mappel::Gauss2DModel, 336
 - mappel::Gauss2DsMAP, 365
 - mappel::Gauss2DsMLE, 397
 - mappel::Gauss2DsModel, 430
- Gauss1DsMAP.cpp, 788
- Gauss1DsMAP.h, 788
- Gauss1DsMAP
 - mappel::Gauss1DsMAP, 195, 196
- Gauss1DsMLE.cpp, 789
- Gauss1DsMLE.h, 790
- Gauss1DsMLE
 - mappel::Gauss1DsMLE, 224
- Gauss1DsModel
 - mappel::Gauss1DsModel, 251
- Gauss1DsModel.cpp, 790
- Gauss1DsModel.h, 791
- Gauss2DModel.cpp, 792
- Gauss2DModel.h, 792
- Gauss2DModel
 - mappel::Gauss2DModel, 279, 280
- Gauss2DModel.cpp, 793
- Gauss2DModel.h, 793
- Gauss2DModel
 - mappel::Gauss2DModel, 309
- Gauss2DModel
 - mappel::Gauss2DModel, 338, 339
- Gauss2DModel.cpp, 794
- Gauss2DModel.h, 795
- Gauss2DsMAP.cpp, 796
- Gauss2DsMAP.h, 796
- Gauss2DsMAP
 - mappel::Gauss2DsMAP, 368
- Gauss2DsMLE.cpp, 797
- Gauss2DsMLE.h, 797
- Gauss2DsMLE
 - mappel::Gauss2DsMLE, 400
- Gauss2DsModel
 - mappel::Gauss2DsModel, 432
- Gauss2DsModel.cpp, 798
- Gauss2DsModel.h, 799
- Gauss2DsxMAP.h, 800
- Gauss2DsxMAP
 - mappel::Gauss2DsxMAP, 462
- Gauss2DsxModel
 - mappel::Gauss2DsxModel, 489
- Gauss2DsxModel.h, 800
- 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, 555
- get_debug_stats
 - mappel::CGaussHeuristicEstimator, 91
 - mappel::CGaussMLE, 97
 - mappel::Estimator, 104
 - mappel::HeuristicEstimator, 510
 - mappel::IterativeMaximizer, 532
 - mappel::NewtonDiagonalMaximizer, 657
 - mappel::NewtonMaximizer, 670
 - mappel::QuasiNewtonMaximizer, 721
 - mappel::SimulatedAnnealingMaximizer, 732
 - mappel::ThreadedEstimator, 762
 - mappel::TrustRegionMaximizer, 773
- get_hyperparam_index
 - mappel::Gauss1DModel, 114
 - mappel::Gauss1DModel, 142
 - mappel::Gauss1DModel, 170
 - mappel::Gauss1DsMAP, 197
 - mappel::Gauss1DsMLE, 226

- mappel::Gauss1DsModel, 253
- mappel::Gauss2DMAP, 282
- mappel::Gauss2DMLE, 312
- mappel::Gauss2DModel, 341
- mappel::Gauss2DsMAP, 371
- mappel::Gauss2DsMLE, 403
- mappel::Gauss2DsModel, 434
- mappel::Gauss2DsxyMAP, 464
- mappel::Gauss2DsxyModel, 491
- mappel::MAPEstimator, 542
- mappel::MCMCAdaptor1Ds, 583
- mappel::MCMCAdaptor1D, 565
- mappel::MCMCAdaptor2Ds, 619
- mappel::MCMCAdaptor2D, 601
- mappel::MLEstimator, 639
- mappel::PointEmitterModel, 685
- get_hyperparam_names
 - mappel::Gauss1DMAP, 115
 - mappel::Gauss1DMLE, 143
 - mappel::Gauss1DModel, 170
 - mappel::Gauss1DsMAP, 198
 - mappel::Gauss1DsMLE, 226
 - mappel::Gauss1DsModel, 253
 - mappel::Gauss2DMAP, 282
 - mappel::Gauss2DMLE, 312
 - mappel::Gauss2DModel, 341
 - mappel::Gauss2DsMAP, 371
 - mappel::Gauss2DsMLE, 403
 - mappel::Gauss2DsModel, 434
 - mappel::Gauss2DsxyMAP, 464
 - mappel::Gauss2DsxyModel, 491
 - mappel::MAPEstimator, 542
 - mappel::MCMCAdaptor1Ds, 583
 - mappel::MCMCAdaptor1D, 565
 - mappel::MCMCAdaptor2Ds, 619
 - mappel::MCMCAdaptor2D, 601
 - mappel::MLEstimator, 640
 - mappel::PointEmitterModel, 685
- get_hyperparam_value
 - mappel::Gauss1DMAP, 115
 - mappel::Gauss1DMLE, 143
 - mappel::Gauss1DModel, 170
 - mappel::Gauss1DsMAP, 198
 - mappel::Gauss1DsMLE, 226
 - mappel::Gauss1DsModel, 253
 - mappel::Gauss2DMAP, 282
 - mappel::Gauss2DMLE, 312
 - mappel::Gauss2DModel, 341
 - mappel::Gauss2DsMAP, 371
 - mappel::Gauss2DsMLE, 403
 - mappel::Gauss2DsModel, 435
 - mappel::Gauss2DsxyMAP, 464
 - mappel::Gauss2DsxyModel, 491
 - mappel::MAPEstimator, 542
- mappel::MCMCAdaptor1Ds, 583
- mappel::MCMCAdaptor1D, 566
- mappel::MCMCAdaptor2Ds, 619
- mappel::MCMCAdaptor2D, 601
- mappel::MLEstimator, 640
- mappel::PointEmitterModel, 685
- get_hyperparams
 - mappel::Gauss1DMAP, 115
 - mappel::Gauss1DMLE, 143
 - mappel::Gauss1DModel, 170
 - mappel::Gauss1DsMAP, 198
 - mappel::Gauss1DsMLE, 226
 - mappel::Gauss1DsModel, 254
 - mappel::Gauss2DMAP, 282
 - mappel::Gauss2DMLE, 312
 - mappel::Gauss2DModel, 341
 - mappel::Gauss2DsMAP, 371
 - mappel::Gauss2DsMLE, 403
 - mappel::Gauss2DsModel, 435
 - mappel::Gauss2DsxyMAP, 464
 - mappel::Gauss2DsxyModel, 491
 - mappel::MAPEstimator, 543
 - mappel::MCMCAdaptor1Ds, 583
 - mappel::MCMCAdaptor1D, 566
 - mappel::MCMCAdaptor2Ds, 619
 - mappel::MCMCAdaptor2D, 601
 - mappel::MLEstimator, 640
 - mappel::PointEmitterModel, 685
- get_image_from_stack
 - mappel::Gauss1DMAP, 115
 - mappel::Gauss1DMLE, 143
 - mappel::Gauss1DModel, 171
 - mappel::Gauss1DsMAP, 198
 - mappel::Gauss1DsMLE, 226
 - mappel::Gauss1DsModel, 254
 - mappel::Gauss2DMAP, 282
 - mappel::Gauss2DMLE, 312
 - mappel::Gauss2DModel, 341
 - mappel::Gauss2DsMAP, 371
 - mappel::Gauss2DsMLE, 403
 - mappel::Gauss2DsModel, 435
 - mappel::Gauss2DsxyMAP, 465
 - mappel::Gauss2DsxyModel, 491
 - mappel::ImageFormat1DBase, 515
 - mappel::ImageFormat2DBase, 522
 - mappel::PoissonNoise1DObjective, 701
 - mappel::PoissonNoise2DObjective, 709
- get_lbound
 - mappel::Gauss1DMAP, 115
 - mappel::Gauss1DMLE, 143
 - mappel::Gauss1DModel, 171
 - mappel::Gauss1DsMAP, 198
 - mappel::Gauss1DsMLE, 226
 - mappel::Gauss1DsModel, 254

- mappel::Gauss2DModel, 341
- mappel::Gauss2DsMAP, 371
- mappel::Gauss2DsMLE, 403
- mappel::Gauss2DsModel, 435
- mappel::Gauss2DsxyMAP, 465
- mappel::Gauss2DsxyModel, 491
- mappel::MAPEstimator, 543
- mappel::MCMCAdaptor1Ds, 584
- mappel::MCMCAdaptor1D, 566
- mappel::MCMCAdaptor2Ds, 620
- mappel::MCMCAdaptor2D, 602
- mappel::MLEstimator, 640
- mappel::PointEmitterModel, 685
- get_max_sigma
 - mappel::Gauss1DsMAP, 198
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 254
 - mappel::Gauss2DsMAP, 371, 372
 - mappel::Gauss2DsMLE, 403, 404
 - mappel::Gauss2DsModel, 435
 - mappel::Gauss2DsxyModel, 491, 492
- get_max_sigma_ratio
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyModel, 492
- get_mcmc_num_phases
 - mappel::Gauss1DModel, 171
 - mappel::Gauss1DsMAP, 198
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 254
 - mappel::Gauss2DModel, 342
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyMAP, 465
 - mappel::MCMCAdaptor1Ds, 584
 - mappel::MCMCAdaptor1D, 566
 - mappel::MCMCAdaptor2Ds, 620
 - mappel::MCMCAdaptor2D, 602
 - mappel::MCMCAdaptorBase, 634
- get_mcmc_sigma_scale
 - mappel::Gauss1DModel, 171
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 254
- mappel::Gauss2DModel, 342
- mappel::Gauss2DsMAP, 372
- mappel::Gauss2DsMLE, 404
- mappel::Gauss2DsModel, 436
- mappel::Gauss2DsxyMAP, 465
- mappel::MCMCAdaptor1Ds, 584
- mappel::MCMCAdaptor1D, 566
- mappel::MCMCAdaptor2Ds, 620
- mappel::MCMCAdaptor2D, 602
- mappel::MCMCAdaptorBase, 634
- get_min_sigma
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 254
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyModel, 492
- get_model
 - mappel::CGaussHeuristicEstimator, 91
 - mappel::CGaussMLE, 98
 - mappel::Estimator, 104
 - mappel::HeuristicEstimator, 510
 - mappel::IterativeMaximizer, 532
 - mappel::NewtonDiagonalMaximizer, 658
 - mappel::NewtonMaximizer, 670
 - mappel::QuasiNewtonMaximizer, 722
 - mappel::SimulatedAnnealingMaximizer, 732
 - mappel::ThreadedEstimator, 762
 - mappel::TrustRegionMaximizer, 773
- get_num_hyperparams
 - mappel::Gauss1DModel, 171
 - mappel::Gauss1DsMAP, 199
 - mappel::Gauss1DsMLE, 227
 - mappel::Gauss1DsModel, 255
 - mappel::Gauss2DModel, 342
 - mappel::Gauss2DsMAP, 372
 - mappel::Gauss2DsMLE, 404
 - mappel::Gauss2DsModel, 436
 - mappel::Gauss2DsxyMAP, 465
 - mappel::Gauss2DsxyModel, 492
 - mappel::MAPEstimator, 543
 - mappel::MCMCAdaptor1Ds, 584
 - mappel::MCMCAdaptor1D, 566
 - mappel::MCMCAdaptor2Ds, 620
 - mappel::MCMCAdaptor2D, 602
 - mappel::MLEstimator, 640
 - mappel::PointEmitterModel, 686

get_num_params

mappel::Gauss1DMAP, 116
 mappel::Gauss1DMLE, 144
 mappel::Gauss1DModel, 171
 mappel::Gauss1DsMAP, 199
 mappel::Gauss1DsMLE, 227
 mappel::Gauss1DsModel, 255
 mappel::Gauss2DMAP, 283
 mappel::Gauss2DMLE, 313
 mappel::Gauss2DModel, 342
 mappel::Gauss2DsMAP, 373
 mappel::Gauss2DsMLE, 405
 mappel::Gauss2DsModel, 436
 mappel::Gauss2DsxyMAP, 465
 mappel::Gauss2DsxyModel, 492
 mappel::MAPEstimator, 543
 mappel::MCMCAdaptor1Ds, 584
 mappel::MCMCAdaptor1D, 566
 mappel::MCMCAdaptor2Ds, 620
 mappel::MCMCAdaptor2D, 602
 mappel::MLEstimator, 640
 mappel::PointEmitterModel, 686

get_num_pixels

mappel::Gauss1DMAP, 116
 mappel::Gauss1DMLE, 144
 mappel::Gauss1DModel, 171
 mappel::Gauss1DsMAP, 199
 mappel::Gauss1DsMLE, 227
 mappel::Gauss1DsModel, 255
 mappel::Gauss2DMAP, 283
 mappel::Gauss2DMLE, 313
 mappel::Gauss2DModel, 342
 mappel::Gauss2DsMAP, 373
 mappel::Gauss2DsMLE, 405
 mappel::Gauss2DsModel, 436
 mappel::Gauss2DsxyMAP, 465
 mappel::Gauss2DsxyModel, 492
 mappel::ImageFormat1DBase, 516
 mappel::ImageFormat2DBase, 522
 mappel::PoissonNoise1DObjective, 702
 mappel::PoissonNoise2DObjective, 709

get_param_names

mappel::Gauss1DMAP, 116
 mappel::Gauss1DMLE, 144
 mappel::Gauss1DModel, 172
 mappel::Gauss1DsMAP, 199
 mappel::Gauss1DsMLE, 228
 mappel::Gauss1DsModel, 255
 mappel::Gauss2DMAP, 283
 mappel::Gauss2DMLE, 313
 mappel::Gauss2DModel, 342
 mappel::Gauss2DsMAP, 373
 mappel::Gauss2DsMLE, 405
 mappel::Gauss2DsModel, 437

mappel::Gauss2DsxyMAP, 466
 mappel::Gauss2DsxyModel, 492
 mappel::MAPEstimator, 543
 mappel::MCMCAdaptor1Ds, 584
 mappel::MCMCAdaptor1D, 567
 mappel::MCMCAdaptor2Ds, 620
 mappel::MCMCAdaptor2D, 602
 mappel::MLEstimator, 640
 mappel::PointEmitterModel, 686

get_prior

mappel::Gauss1DMAP, 116
 mappel::Gauss1DMLE, 144
 mappel::Gauss1DModel, 172
 mappel::Gauss1DsMAP, 199, 200
 mappel::Gauss1DsMLE, 228
 mappel::Gauss1DsModel, 255
 mappel::Gauss2DMAP, 283
 mappel::Gauss2DMLE, 313
 mappel::Gauss2DModel, 342, 343
 mappel::Gauss2DsMAP, 373
 mappel::Gauss2DsMLE, 405
 mappel::Gauss2DsModel, 437
 mappel::Gauss2DsxyMAP, 466
 mappel::Gauss2DsxyModel, 493
 mappel::MAPEstimator, 543
 mappel::MCMCAdaptor1Ds, 584, 585
 mappel::MCMCAdaptor1D, 567
 mappel::MCMCAdaptor2Ds, 620, 621
 mappel::MCMCAdaptor2D, 602, 603
 mappel::MLEstimator, 641
 mappel::PointEmitterModel, 686

get_psf_sigma

mappel::Gauss1DMAP, 116, 117
 mappel::Gauss1DMLE, 144, 145
 mappel::Gauss1DModel, 172
 mappel::Gauss2DMAP, 284
 mappel::Gauss2DMLE, 314
 mappel::Gauss2DModel, 343
 mappel::Gauss2DsxyMAP, 466

get_rng_generator

mappel::Gauss1DMAP, 117
 mappel::Gauss1DMLE, 145
 mappel::Gauss1DModel, 172
 mappel::Gauss1DsMAP, 200
 mappel::Gauss1DsMLE, 228
 mappel::Gauss1DsModel, 255
 mappel::Gauss2DMAP, 284
 mappel::Gauss2DMLE, 314
 mappel::Gauss2DModel, 343
 mappel::Gauss2DsMAP, 373
 mappel::Gauss2DsMLE, 405
 mappel::Gauss2DsModel, 437
 mappel::Gauss2DsxyMAP, 466
 mappel::Gauss2DsxyModel, 493

- mappel::MAPEstimator, [544](#)
- mappel::MCMCAdaptor1Ds, [585](#)
- mappel::MCMCAdaptor1D, [567](#)
- mappel::MCMCAdaptor2Ds, [621](#)
- mappel::MCMCAdaptor2D, [603](#)
- mappel::MLEstimator, [641](#)
- mappel::PointEmitterModel, [686](#)
- get_rng_manager
 - mappel::Gauss1DMap, [117](#)
 - mappel::Gauss1DMLE, [145](#)
 - mappel::Gauss1DModel, [172](#)
 - mappel::Gauss1DsMap, [200](#)
 - mappel::Gauss1DsMLE, [228](#)
 - mappel::Gauss1DsModel, [256](#)
 - mappel::Gauss2DMap, [284](#)
 - mappel::Gauss2DMLE, [314](#)
 - mappel::Gauss2DModel, [343](#)
 - mappel::Gauss2DsMap, [373](#)
 - mappel::Gauss2DsMLE, [405](#)
 - mappel::Gauss2DsModel, [437](#)
 - mappel::Gauss2DsxyMap, [466](#)
 - mappel::Gauss2DsxyModel, [493](#)
 - mappel::MAPEstimator, [544](#)
 - mappel::MCMCAdaptor1Ds, [585](#)
 - mappel::MCMCAdaptor1D, [567](#)
 - mappel::MCMCAdaptor2Ds, [621](#)
 - mappel::MCMCAdaptor2D, [603](#)
 - mappel::MLEstimator, [641](#)
 - mappel::PointEmitterModel, [686](#)
- get_size
 - 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, [374](#)
 - mappel::Gauss2DsMLE, [406](#)
 - mappel::Gauss2DsModel, [437](#)
 - mappel::Gauss2DsxyMap, [467](#)
 - mappel::Gauss2DsxyModel, [493](#)
 - mappel::ImageFormat1DBase, [516](#)
 - mappel::ImageFormat2DBase, [522](#)
 - mappel::PoissonNoise1DObjective, [702](#)
 - mappel::PoissonNoise2DObjective, [709](#)
- get_size_image_stack
 - mappel::Gauss1DMap, [117](#)
 - mappel::Gauss1DMLE, [145](#)
 - mappel::Gauss1DModel, [173](#)
 - mappel::Gauss1DsMap, [200](#)
 - mappel::Gauss1DsMLE, [229](#)
- mappel::Gauss1DsModel, [256](#)
- mappel::Gauss2DMap, [284](#)
- mappel::Gauss2DMLE, [314](#)
- mappel::Gauss2DModel, [344](#)
- mappel::Gauss2DsMap, [374](#)
- mappel::Gauss2DsMLE, [406](#)
- mappel::Gauss2DsModel, [438](#)
- mappel::Gauss2DsxyMap, [467](#)
- mappel::Gauss2DsxyModel, [493](#)
- mappel::ImageFormat1DBase, [516](#)
- mappel::ImageFormat2DBase, [523](#)
- mappel::PoissonNoise1DObjective, [702](#)
- mappel::PoissonNoise2DObjective, [709](#)
- get_stats
 - mappel::CGaussHeuristicEstimator, [91](#)
 - mappel::CGaussMLE, [98](#)
 - mappel::Estimator, [104](#)
 - mappel::Gauss1DMap, [117](#)
 - mappel::Gauss1DMLE, [145](#)
 - mappel::Gauss1DModel, [173](#)
 - mappel::Gauss1DsMap, [200](#)
 - mappel::Gauss1DsMLE, [229](#)
 - mappel::Gauss1DsModel, [256](#)
 - mappel::Gauss2DMap, [285](#)
 - mappel::Gauss2DMLE, [315](#)
 - mappel::Gauss2DModel, [344](#)
 - mappel::Gauss2DsMap, [374](#)
 - mappel::Gauss2DsMLE, [406](#)
 - mappel::Gauss2DsModel, [438](#)
 - mappel::Gauss2DsxyMap, [467](#)
 - mappel::Gauss2DsxyModel, [494](#)
 - mappel::HeuristicEstimator, [510](#)
 - mappel::ImageFormat1DBase, [516](#)
 - mappel::ImageFormat2DBase, [523](#)
 - mappel::IterativeMaximizer, [532](#)
 - mappel::MAPEstimator, [544](#)
 - mappel::MCMCAdaptor1Ds, [585](#)
 - mappel::MCMCAdaptor1D, [567](#)
 - mappel::MCMCAdaptor2Ds, [621](#)
 - mappel::MCMCAdaptor2D, [603](#)
 - mappel::MCMCAdaptorBase, [634](#)
 - mappel::MLEstimator, [641](#)
 - mappel::NewtonDiagonalMaximizer, [658](#)
 - mappel::NewtonMaximizer, [670](#)
 - mappel::PointEmitterModel, [687](#)
 - mappel::PoissonNoise1DObjective, [702](#)
 - mappel::PoissonNoise2DObjective, [709](#)
 - mappel::QuasiNewtonMaximizer, [722](#)
 - mappel::SimulatedAnnealingMaximizer, [732](#)
 - mappel::ThreadedEstimator, [763](#)
 - mappel::TrustRegionMaximizer, [773](#)
- get_theta_sequence
 - mappel::IterativeMaximizer::MaximizerData, [555](#)
- get_theta_sequence_rllh

- mappel::IterativeMaximizer::MaximizerData, 555
- get_ubound
 - mappel::Gauss1DMAP, 117
 - mappel::Gauss1DMLE, 145
 - mappel::Gauss1DModel, 173
 - mappel::Gauss1DsMAP, 201
 - mappel::Gauss1DsMLE, 229
 - mappel::Gauss1DsModel, 256
 - mappel::Gauss2DMAP, 285
 - mappel::Gauss2DMLE, 315
 - mappel::Gauss2DModel, 344
 - mappel::Gauss2DsMAP, 374
 - mappel::Gauss2DsMLE, 406
 - mappel::Gauss2DsModel, 438
 - mappel::Gauss2DsxyMAP, 467
 - mappel::Gauss2DsxyModel, 494
 - mappel::MAPEstimator, 544
 - mappel::MCMCAdaptor1Ds, 585
 - mappel::MCMCAdaptor1D, 567
 - mappel::MCMCAdaptor2Ds, 621
 - mappel::MCMCAdaptor2D, 603
 - mappel::MLEstimator, 641
 - mappel::PointEmitterModel, 687
- getIteration
 - mappel::IterativeMaximizer::MaximizerData, 555
- global_default_mcmc_sigma_scale
 - mappel::Gauss1DMAP, 130
 - mappel::Gauss1DMLE, 158
 - mappel::Gauss1DModel, 185
 - mappel::Gauss1DsMAP, 214
 - mappel::Gauss1DsMLE, 242
 - mappel::Gauss1DsModel, 269
 - mappel::Gauss2DMAP, 298
 - mappel::Gauss2DMLE, 328
 - mappel::Gauss2DModel, 356
 - mappel::Gauss2DsMAP, 388
 - mappel::Gauss2DsMLE, 420
 - mappel::Gauss2DsModel, 451
 - mappel::Gauss2DsxyMAP, 479
 - mappel::MCMCAdaptor1Ds, 594
 - mappel::MCMCAdaptor1D, 576
 - mappel::MCMCAdaptor2Ds, 630
 - mappel::MCMCAdaptor2D, 612
 - mappel::MCMCAdaptorBase, 635
- global_max_mcmc_sigma_scale
 - mappel::Gauss1DMAP, 130
 - mappel::Gauss1DMLE, 158
 - mappel::Gauss1DModel, 185
 - 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, 479
- mappel::MCMCAdaptor1Ds, 594
- mappel::MCMCAdaptor1D, 576
- mappel::MCMCAdaptor2Ds, 630
- mappel::MCMCAdaptor2D, 612
- mappel::MCMCAdaptorBase, 635
- global_max_psf_sigma
 - mappel::Gauss1DMAP, 130
 - mappel::Gauss1DMLE, 158
 - mappel::Gauss1DModel, 185
 - mappel::Gauss1DsMAP, 214
 - mappel::Gauss1DsMLE, 242
 - mappel::Gauss1DsModel, 269
 - mappel::Gauss2DMAP, 298
 - mappel::Gauss2DMLE, 328
 - mappel::Gauss2DModel, 357
 - mappel::Gauss2DsMAP, 389
 - mappel::Gauss2DsMLE, 421
 - mappel::Gauss2DsModel, 452
 - mappel::Gauss2DsxyMAP, 479
 - mappel::Gauss2DsxyModel, 502
 - mappel::MAPEstimator, 551
 - mappel::MCMCAdaptor1Ds, 594
 - mappel::MCMCAdaptor1D, 576
 - mappel::MCMCAdaptor2Ds, 630
 - mappel::MCMCAdaptor2D, 612
 - mappel::MLEstimator, 648
 - mappel::PointEmitterModel, 693
- global_max_size
 - 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, 389
 - mappel::Gauss2DsMLE, 421
 - mappel::Gauss2DsModel, 452
 - mappel::Gauss2DsxyMAP, 480
 - mappel::Gauss2DsxyModel, 502
 - mappel::ImageFormat1DBase, 517
 - mappel::ImageFormat2DBase, 524
 - mappel::PoissonNoise1DObjective, 703
 - mappel::PoissonNoise2DObjective, 711
- global_min_psf_sigma
 - 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, 389
- mappel::Gauss2DsMLE, 421
- mappel::Gauss2DsModel, 452
- mappel::Gauss2DsxyMAP, 480
- mappel::Gauss2DsxyModel, 503
- mappel::MAPEstimator, 551
- mappel::MCMCAdaptor1Ds, 594
- mappel::MCMCAdaptor1D, 576
- mappel::MCMCAdaptor2Ds, 630
- mappel::MCMCAdaptor2D, 612
- mappel::MLEstimator, 648
- mappel::PointEmitterModel, 694
- global_min_size
 - mappel::Gauss1DMAP, 131
 - mappel::Gauss1DMLE, 159
 - mappel::Gauss1DModel, 186
 - mappel::Gauss1DsMAP, 214
 - 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, 503
 - mappel::ImageFormat1DBase, 517
 - mappel::ImageFormat2DBase, 524
 - mappel::PoissonNoise1DObjective, 704
 - mappel::PoissonNoise2DObjective, 711
- grad
 - mappel::IterativeMaximizer::MaximizerData, 558
 - 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::IterativeMaximizer, 528
 - mappel::NewtonDiagonalMaximizer, 653
 - mappel::NewtonMaximizer, 666
 - mappel::QuasiNewtonMaximizer, 717
- mappel::TrustRegionMaximizer, 768
- Gx
 - mappel::Gauss1DModel::Stencil, 737
 - mappel::Gauss1DsModel::Stencil, 741
 - mappel::Gauss2DModel::Stencil, 757
 - mappel::Gauss2DsModel::Stencil, 752
 - mappel::Gauss2DsxyModel::Stencil, 746
- Gy
 - mappel::Gauss2DModel::Stencil, 757
 - mappel::Gauss2DsModel::Stencil, 752
 - mappel::Gauss2DsxyModel::Stencil, 746
- has_fixed_parameters
 - mappel::IterativeMaximizer::MaximizerData, 558
- has_hyperparam
 - mappel::Gauss1DMAP, 118
 - mappel::Gauss1DMLE, 146
 - mappel::Gauss1DModel, 173
 - mappel::Gauss1DsMAP, 201
 - mappel::Gauss1DsMLE, 229
 - mappel::Gauss1DsModel, 256
 - mappel::Gauss2DMAP, 285
 - mappel::Gauss2DMLE, 315
 - mappel::Gauss2DModel, 344
 - mappel::Gauss2DsMAP, 374
 - mappel::Gauss2DsMLE, 406
 - mappel::Gauss2DsModel, 438
 - mappel::Gauss2DsxyMAP, 467
 - mappel::Gauss2DsxyModel, 494
 - mappel::MAPEstimator, 544
 - mappel::MCMCAdaptor1Ds, 585
 - mappel::MCMCAdaptor1D, 568
 - mappel::MCMCAdaptor2Ds, 621
 - mappel::MCMCAdaptor2D, 603
 - mappel::MLEstimator, 641
 - mappel::PointEmitterModel, 687
- 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, 507
- I
 - mappel::Gauss1DModel::Stencil, 736
 - mappel::Gauss1DsModel::Stencil, 740
 - mappel::Gauss2DModel::Stencil, 755
 - mappel::Gauss2DsModel::Stencil, 749
 - mappel::Gauss2DsxyModel::Stencil, 744
- icontains
 - mappel, 35

- IdxMatT
 - mappel, [25](#)
- IdxVecT
 - mappel, [25](#)
- IdxT
 - mappel, [25](#)
- im
 - mappel::IterativeMaximizer::MaximizerData, [558](#)
- ImageCoordT
 - mappel, [25](#)
 - mappel::Gauss1DMap, [110](#)
 - mappel::Gauss1DMLE, [138](#)
 - mappel::Gauss1DModel, [166](#)
 - mappel::Gauss1DsMap, [193](#)
 - mappel::Gauss1DsMLE, [222](#)
 - mappel::Gauss1DsModel, [249](#)
 - mappel::Gauss2DMap, [277](#)
 - mappel::Gauss2DMLE, [307](#)
 - mappel::Gauss2DModel, [336](#)
 - mappel::Gauss2DsMap, [365](#)
 - mappel::Gauss2DsMLE, [397](#)
 - mappel::Gauss2DsModel, [430](#)
 - mappel::Gauss2DsxyMap, [460](#)
 - mappel::Gauss2DsxyModel, [487](#)
 - mappel::ImageFormat1DBase, [514](#)
 - mappel::ImageFormat2DBase, [520](#)
 - mappel::PoissonNoise1DObjective, [699](#)
 - mappel::PoissonNoise2DObjective, [706](#)
- ImageFormat1DBase
 - mappel::ImageFormat1DBase, [515](#)
- ImageFormat1DBase.cpp, [801](#)
- ImageFormat1DBase.h, [802](#)
- ImageFormat2DBase
 - mappel::ImageFormat2DBase, [521](#)
- ImageFormat2DBase.cpp, [803](#)
- ImageFormat2DBase.h, [803](#)
- ImagePixelT
 - mappel, [26](#)
 - mappel::Gauss1DMap, [110](#)
 - mappel::Gauss1DMLE, [138](#)
 - mappel::Gauss1DModel, [166](#)
 - mappel::Gauss1DsMap, [193](#)
 - mappel::Gauss1DsMLE, [222](#)
 - mappel::Gauss1DsModel, [249](#)
 - mappel::Gauss2DMap, [277](#)
 - mappel::Gauss2DMLE, [307](#)
 - mappel::Gauss2DModel, [337](#)
 - mappel::Gauss2DsMap, [366](#)
 - mappel::Gauss2DsMLE, [398](#)
 - mappel::Gauss2DsModel, [430](#)
 - mappel::Gauss2DsxyMap, [460](#)
 - mappel::Gauss2DsxyModel, [487](#)
 - mappel::ImageFormat1DBase, [514](#)
 - mappel::ImageFormat2DBase, [520](#)
- mappel::PoissonNoise1DObjective, [699](#)
- mappel::PoissonNoise2DObjective, [706](#)
- ImageShapeT
 - mappel::Gauss1DMap, [110](#)
 - mappel::Gauss1DMLE, [138](#)
 - mappel::Gauss1DModel, [166](#)
 - mappel::Gauss1DsMap, [193](#)
 - mappel::Gauss1DsMLE, [222](#)
 - mappel::Gauss1DsModel, [250](#)
 - mappel::Gauss2DMap, [277](#)
 - mappel::Gauss2DMLE, [307](#)
 - mappel::Gauss2DModel, [337](#)
 - mappel::Gauss2DsMap, [366](#)
 - mappel::Gauss2DsMLE, [398](#)
 - mappel::Gauss2DsModel, [430](#)
 - mappel::Gauss2DsxyMap, [460](#)
 - mappel::Gauss2DsxyModel, [487](#)
 - mappel::ImageFormat1DBase, [514](#)
 - mappel::ImageFormat2DBase, [520](#)
 - mappel::PoissonNoise1DObjective, [699](#)
 - mappel::PoissonNoise2DObjective, [706](#)
- ImageSizeShapeT
 - mappel::Gauss1DMap, [110](#)
 - mappel::Gauss1DMLE, [138](#)
 - mappel::Gauss1DModel, [167](#)
 - mappel::Gauss1DsMap, [193](#)
 - mappel::Gauss1DsMLE, [222](#)
 - mappel::Gauss1DsModel, [250](#)
 - mappel::Gauss2DMap, [277](#)
 - mappel::Gauss2DMLE, [307](#)
 - mappel::Gauss2DModel, [337](#)
 - mappel::Gauss2DsMap, [366](#)
 - mappel::Gauss2DsMLE, [398](#)
 - mappel::Gauss2DsModel, [430](#)
 - mappel::Gauss2DsxyMap, [460](#)
 - mappel::Gauss2DsxyModel, [487](#)
 - mappel::ImageFormat1DBase, [514](#)
 - mappel::ImageFormat2DBase, [520](#)
 - mappel::PoissonNoise1DObjective, [699](#)
 - mappel::PoissonNoise2DObjective, [707](#)
- ImageSizeVecShapeT
 - mappel::Gauss1DMap, [111](#)
 - mappel::Gauss1DMLE, [139](#)
 - mappel::Gauss1DModel, [167](#)
 - mappel::Gauss1DsMap, [194](#)
 - mappel::Gauss1DsMLE, [222](#)
 - mappel::Gauss1DsModel, [250](#)
 - mappel::Gauss2DMap, [277](#)
 - mappel::Gauss2DMLE, [307](#)
 - mappel::Gauss2DModel, [337](#)
 - mappel::Gauss2DsMap, [366](#)
 - mappel::Gauss2DsMLE, [398](#)
 - mappel::Gauss2DsModel, [430](#)
 - mappel::Gauss2DsxyMap, [461](#)

- mappel::Gauss2DsxyModel, [487](#)
- mappel::ImageFormat1DBase, [514](#)
- mappel::ImageFormat2DBase, [520](#)
- mappel::PoissonNoise1DObjective, [700](#)
- mappel::PoissonNoise2DObjective, [707](#)
- ImageSizeVecT
 - mappel::Gauss1DMap, [111](#)
 - mappel::Gauss1DMLE, [139](#)
 - mappel::Gauss1DModel, [167](#)
 - mappel::Gauss1DsMAP, [194](#)
 - mappel::Gauss1DsMLE, [222](#)
 - mappel::Gauss1DsModel, [250](#)
 - mappel::Gauss2DMap, [278](#)
 - mappel::Gauss2DMLE, [308](#)
 - mappel::Gauss2DModel, [337](#)
 - mappel::Gauss2DsMAP, [366](#)
 - mappel::Gauss2DsMLE, [398](#)
 - mappel::Gauss2DsModel, [431](#)
 - mappel::Gauss2DsxyMAP, [461](#)
 - mappel::Gauss2DsxyModel, [488](#)
 - mappel::ImageFormat1DBase, [514](#)
 - mappel::ImageFormat2DBase, [520](#)
 - mappel::PoissonNoise1DObjective, [700](#)
 - mappel::PoissonNoise2DObjective, [707](#)
- ImageSizeT
 - mappel::Gauss1DMap, [111](#)
 - mappel::Gauss1DMLE, [139](#)
 - mappel::Gauss1DModel, [167](#)
 - mappel::Gauss1DsMAP, [194](#)
 - mappel::Gauss1DsMLE, [222](#)
 - mappel::Gauss1DsModel, [250](#)
 - mappel::Gauss2DMap, [277](#)
 - mappel::Gauss2DMLE, [307](#)
 - mappel::Gauss2DModel, [337](#)
 - mappel::Gauss2DsMAP, [366](#)
 - mappel::Gauss2DsMLE, [398](#)
 - mappel::Gauss2DsModel, [430](#)
 - mappel::Gauss2DsxyMAP, [461](#)
 - mappel::Gauss2DsxyModel, [487](#)
 - mappel::ImageFormat1DBase, [514](#)
 - mappel::ImageFormat2DBase, [520](#)
 - mappel::PoissonNoise1DObjective, [699](#)
 - mappel::PoissonNoise2DObjective, [707](#)
- ImageStackShapeT
 - mappel::Gauss1DMap, [111](#)
 - mappel::Gauss1DMLE, [139](#)
 - mappel::Gauss1DModel, [167](#)
 - mappel::Gauss1DsMAP, [194](#)
 - mappel::Gauss1DsMLE, [222](#)
 - mappel::Gauss1DsModel, [250](#)
 - mappel::Gauss2DMap, [278](#)
 - mappel::Gauss2DMLE, [308](#)
 - mappel::Gauss2DModel, [337](#)
 - mappel::Gauss2DsMAP, [366](#)
- mappel::Gauss2DsMLE, [398](#)
- mappel::Gauss2DsModel, [431](#)
- mappel::Gauss2DsxyMAP, [461](#)
- mappel::Gauss2DsxyModel, [488](#)
- mappel::ImageFormat1DBase, [514](#)
- mappel::ImageFormat2DBase, [521](#)
- mappel::PoissonNoise1DObjective, [700](#)
- mappel::PoissonNoise2DObjective, [707](#)
- ImageStackT
 - mappel, [26](#)
 - mappel::Gauss1DMap, [111](#)
 - mappel::Gauss1DMLE, [139](#)
 - mappel::Gauss1DModel, [167](#)
 - mappel::Gauss1DsMAP, [194](#)
 - mappel::Gauss1DsMLE, [223](#)
 - mappel::Gauss1DsModel, [250](#)
 - mappel::Gauss2DMap, [278](#)
 - mappel::Gauss2DMLE, [308](#)
 - mappel::Gauss2DModel, [338](#)
 - mappel::Gauss2DsMAP, [367](#)
 - mappel::Gauss2DsMLE, [399](#)
 - mappel::Gauss2DsModel, [431](#)
 - mappel::Gauss2DsxyMAP, [461](#)
 - mappel::Gauss2DsxyModel, [488](#)
 - mappel::ImageFormat1DBase, [514](#)
 - mappel::ImageFormat2DBase, [521](#)
 - mappel::PoissonNoise1DObjective, [700](#)
 - mappel::PoissonNoise2DObjective, [707](#)
- ImageT
 - mappel, [26](#)
 - mappel::Gauss1DMap, [111](#)
 - mappel::Gauss1DMLE, [139](#)
 - mappel::Gauss1DModel, [167](#)
 - mappel::Gauss1DsMAP, [194](#)
 - mappel::Gauss1DsMLE, [223](#)
 - mappel::Gauss1DsModel, [251](#)
 - mappel::Gauss2DMap, [278](#)
 - mappel::Gauss2DMLE, [308](#)
 - mappel::Gauss2DModel, [338](#)
 - mappel::Gauss2DsMAP, [367](#)
 - mappel::Gauss2DsMLE, [399](#)
 - mappel::Gauss2DsModel, [431](#)
 - mappel::Gauss2DsxyMAP, [461](#)
 - mappel::Gauss2DsxyModel, [488](#)
 - mappel::ImageFormat1DBase, [515](#)
 - mappel::ImageFormat2DBase, [521](#)
 - mappel::PoissonNoise1DObjective, [700](#)
 - mappel::PoissonNoise2DObjective, [707](#)
- initial_theta_estimate
 - mappel::Gauss1DMap, [118](#)
 - mappel::Gauss1DMLE, [146](#)
 - mappel::Gauss1DModel, [173](#), [174](#)
 - mappel::Gauss1DsMAP, [201](#)
 - mappel::Gauss1DsMLE, [229](#)

- mappel::Gauss1DsModel, [257](#)
- mappel::Gauss2DMAP, [285](#)
- mappel::Gauss2DMLE, [315](#)
- mappel::Gauss2DModel, [344](#), [345](#)
- mappel::Gauss2DsMAP, [374](#), [375](#)
- mappel::Gauss2DsMLE, [406](#), [407](#)
- mappel::Gauss2DsModel, [438](#), [439](#)
- mappel::Gauss2DsxyMAP, [467](#), [468](#)
- mappel::Gauss2DsxyModel, [494](#)
- Install.md, [804](#)
- is_negative_definite
 - mappel, [35](#)
- is_positive_definite
 - mappel, [35](#)
- is_symmetric
 - mappel, [35](#)
- istarts_with
 - mappel, [35](#), [36](#)
- IterativeMaximizer
 - mappel::IterativeMaximizer, [528](#)
- lambda_min
 - mappel::IterativeMaximizer, [534](#)
 - mappel::NewtonDiagonalMaximizer, [660](#)
 - mappel::NewtonMaximizer, [673](#)
 - mappel::QuasiNewtonMaximizer, [724](#)
 - mappel::TrustRegionMaximizer, [777](#)
- lambda_term_color
 - mappel, [36](#)
- last_backtrack_idx
 - mappel::IterativeMaximizer, [535](#)
 - mappel::NewtonDiagonalMaximizer, [661](#)
 - mappel::NewtonMaximizer, [673](#)
 - mappel::QuasiNewtonMaximizer, [725](#)
 - mappel::TrustRegionMaximizer, [777](#)
- lbound
 - 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, [357](#)
 - mappel::Gauss2DsMAP, [389](#)
 - mappel::Gauss2DsMLE, [421](#)
 - mappel::Gauss2DsModel, [452](#)
 - mappel::Gauss2DsxyMAP, [480](#)
 - mappel::Gauss2DsxyModel, [503](#)
 - mappel::IterativeMaximizer::MaximizerData, [558](#)
 - mappel::MAPEstimator, [551](#)
 - mappel::MCMCAdaptor1Ds, [594](#)
 - mappel::MCMCAdaptor1D, [576](#)
 - mappel::MCMCAdaptor2Ds, [631](#)
 - mappel::MCMCAdaptor2D, [612](#)
 - mappel::MLEstimator, [648](#)
 - mappel::PointEmitterModel, [694](#)
- 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, [532](#)
 - mappel::NewtonDiagonalMaximizer, [658](#)
 - mappel::NewtonMaximizer, [670](#)
 - mappel::QuasiNewtonMaximizer, [722](#)
 - mappel::TrustRegionMaximizer, [773](#)
- log_likelihood
 - PoissonGaussianNoise2DObjective.h, [829](#)
- log_likelihood_stack
 - mappel, [36](#)
- log_prior_l_const
 - mappel::PriorMAP1DObjective, [714](#)
- log_prior_beta2_const
 - mappel, [37](#)
- log_prior_beta_const
 - mappel, [37](#)
- log_prior_bg_const
 - mappel::PriorMAP1DObjective, [714](#)
- log_prior_gamma_const
 - mappel, [37](#)
- log_prior_normal_const
 - mappel, [37](#)
- log_prior_pareto_const
 - mappel, [37](#)
- log_prior_pos_const
 - mappel::PriorMAP1DObjective, [714](#)
- LogicalError
 - mappel::LogicalError, [538](#)
- MAPEstimator
 - mappel::MAPEstimator, [541](#)
- MAPEstimator.h, [804](#)
- MCMCAdaptor1D.cpp, [807](#)
- MCMCAdaptor1D.h, [807](#)
- MCMCAdaptor1Ds
 - mappel::MCMCAdaptor1Ds, [581](#), [582](#)
- MCMCAdaptor1Ds.cpp, [808](#)
- MCMCAdaptor1Ds.h, [809](#)
- MCMCAdaptor1D
 - mappel::MCMCAdaptor1D, [564](#)

- MCMCAdaptor2D.cpp, [809](#)
- MCMCAdaptor2D.h, [810](#)
- MCMCAdaptor2Ds
 - mappel::MCMCAdaptor2Ds, [617](#), [618](#)
- MCMCAdaptor2Ds.cpp, [810](#)
- MCMCAdaptor2Ds.h, [811](#)
- MCMCAdaptor2D
 - mappel::MCMCAdaptor2D, [599](#), [600](#)
- MCMCAdaptorBase
 - mappel::MCMCAdaptorBase, [633](#)
- MCMCAdaptorBase.cpp, [811](#)
- MCMCAdaptorBase.h, [812](#)
- MLEstimator
 - mappel::MLEstimator, [638](#)
- MLEstimator.h, [812](#)
- 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, [118](#)
 - mappel::Gauss1DMLE, [146](#)
 - mappel::Gauss1DModel, [174](#)
 - mappel::Gauss1DsMAP, [201](#)
 - mappel::Gauss1DsMLE, [229](#)
 - mappel::Gauss1DsModel, [257](#)
 - mappel::Gauss2DMAP, [286](#)
 - mappel::Gauss2DMLE, [316](#)
 - mappel::Gauss2DModel, [345](#)
 - mappel::Gauss2DsMAP, [375](#)
 - mappel::Gauss2DsMLE, [407](#)
 - mappel::Gauss2DsModel, [439](#)
 - mappel::Gauss2DsxyMAP, [468](#)
 - mappel::Gauss2DsxyModel, [494](#)
- make_default_prior_beta_position
 - mappel::Gauss1DMAP, [118](#)
 - mappel::Gauss1DMLE, [146](#)
 - mappel::Gauss1DModel, [174](#)
 - mappel::Gauss1DsMAP, [201](#)
 - mappel::Gauss1DsMLE, [230](#)
 - mappel::Gauss1DsModel, [257](#)
 - mappel::Gauss2DMAP, [286](#)
 - mappel::Gauss2DMLE, [316](#)
 - mappel::Gauss2DModel, [345](#)
- mappel::Gauss2DsMAP, [375](#)
- mappel::Gauss2DsMLE, [407](#)
- mappel::Gauss2DsModel, [439](#)
- mappel::Gauss2DsxyMAP, [468](#)
- mappel::Gauss2DsxyModel, [494](#)
- make_estimator
 - mappel::methods, [57](#), [58](#)
 - PoissonGaussianNoise2DObjective.h, [829](#)
- make_gaussian_stencil
 - mappel, [39](#)
- make_image
 - mappel::Gauss1DMAP, [119](#)
 - mappel::Gauss1DMLE, [147](#)
 - mappel::Gauss1DModel, [174](#)
 - mappel::Gauss1DsMAP, [202](#)
 - mappel::Gauss1DsMLE, [230](#)
 - mappel::Gauss1DsModel, [257](#)
 - mappel::Gauss2DMAP, [286](#)
 - mappel::Gauss2DMLE, [316](#)
 - mappel::Gauss2DModel, [345](#)
 - mappel::Gauss2DsMAP, [376](#)
 - mappel::Gauss2DsMLE, [408](#)
 - mappel::Gauss2DsModel, [439](#)
 - mappel::Gauss2DsxyMAP, [468](#)
 - mappel::Gauss2DsxyModel, [494](#)
 - mappel::ImageFormat1DBase, [516](#)
 - mappel::ImageFormat2DBase, [523](#)
 - mappel::PoissonNoise1DObjective, [702](#)
 - mappel::PoissonNoise2DObjective, [710](#)
- make_image_stack
 - mappel::Gauss1DMAP, [119](#)
 - mappel::Gauss1DMLE, [147](#)
 - mappel::Gauss1DModel, [174](#)
 - mappel::Gauss1DsMAP, [202](#)
 - mappel::Gauss1DsMLE, [230](#)
 - mappel::Gauss1DsModel, [258](#)
 - mappel::Gauss2DMAP, [286](#)
 - mappel::Gauss2DMLE, [316](#)
 - mappel::Gauss2DModel, [345](#)
 - mappel::Gauss2DsMAP, [376](#)
 - mappel::Gauss2DsMLE, [408](#)

- mappel::Gauss2DsModel, [440](#)
- mappel::Gauss2DsxyMAP, [468](#)
- mappel::Gauss2DsxyModel, [494](#)
- mappel::ImageFormat1DBase, [516](#)
- mappel::ImageFormat2DBase, [523](#)
- mappel::PoissonNoise1DObjective, [702](#)
- mappel::PoissonNoise2DObjective, [710](#)
- make_internal_1Dsum_estimator
 - mappel::Gauss2DMAP, [286](#)
 - mappel::Gauss2DMLE, [316](#)
 - mappel::Gauss2DModel, [346](#)
 - mappel::Gauss2DsMAP, [376](#)
 - mappel::Gauss2DsMLE, [408](#)
 - mappel::Gauss2DsModel, [440](#)
- make_param
 - mappel::Gauss1DMAP, [119](#)
 - mappel::Gauss1DMLE, [147](#)
 - mappel::Gauss1DModel, [174](#), [175](#)
 - mappel::Gauss1DsMAP, [202](#)
 - mappel::Gauss1DsMLE, [230](#)
 - 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, [495](#)
 - mappel::MAPEstimator, [544](#), [545](#)
 - mappel::MCMCAdaptor1Ds, [585](#), [586](#)
 - mappel::MCMCAdaptor1D, [568](#)
 - mappel::MCMCAdaptor2Ds, [621](#), [622](#)
 - mappel::MCMCAdaptor2D, [603](#), [604](#)
 - mappel::MLEstimator, [642](#)
 - mappel::PointEmitterModel, [687](#)
- make_param_mat
 - mappel::Gauss1DMAP, [119](#)
 - mappel::Gauss1DMLE, [147](#)
 - mappel::Gauss1DModel, [175](#)
 - mappel::Gauss1DsMAP, [202](#), [203](#)
 - mappel::Gauss1DsMLE, [231](#)
 - mappel::Gauss1DsModel, [258](#)
 - mappel::Gauss2DMAP, [287](#)
 - mappel::Gauss2DMLE, [317](#)
 - mappel::Gauss2DModel, [346](#)
 - mappel::Gauss2DsMAP, [377](#)
 - mappel::Gauss2DsMLE, [409](#)
 - mappel::Gauss2DsModel, [440](#)
 - mappel::Gauss2DsxyMAP, [469](#)
 - mappel::Gauss2DsxyModel, [495](#)
 - mappel::MAPEstimator, [545](#)
 - mappel::MCMCAdaptor1Ds, [586](#)
 - mappel::MCMCAdaptor1D, [568](#)
- mappel::MCMCAdaptor2Ds, [622](#)
- mappel::MCMCAdaptor2D, [604](#)
- mappel::MLEstimator, [642](#)
- mappel::PointEmitterModel, [687](#), [688](#)
- make_param_mat_stack
 - 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](#), [347](#)
 - mappel::Gauss2DsMAP, [377](#)
 - mappel::Gauss2DsMLE, [409](#)
 - mappel::Gauss2DsModel, [441](#)
 - mappel::Gauss2DsxyMAP, [469](#)
 - mappel::Gauss2DsxyModel, [495](#)
 - mappel::MAPEstimator, [545](#)
 - mappel::MCMCAdaptor1Ds, [586](#)
 - mappel::MCMCAdaptor1D, [568](#), [569](#)
 - mappel::MCMCAdaptor2Ds, [622](#)
 - mappel::MCMCAdaptor2D, [604](#)
 - mappel::MLEstimator, [642](#)
 - mappel::PointEmitterModel, [688](#)
- make_param_stack
 - mappel::Gauss1DMAP, [120](#)
 - mappel::Gauss1DMLE, [148](#)
 - mappel::Gauss1DModel, [175](#)
 - mappel::Gauss1DsMAP, [203](#)
 - mappel::Gauss1DsMLE, [231](#)
 - mappel::Gauss1DsModel, [259](#)
 - mappel::Gauss2DMAP, [288](#)
 - mappel::Gauss2DMLE, [318](#)
 - mappel::Gauss2DModel, [347](#)
 - mappel::Gauss2DsMAP, [377](#)
 - mappel::Gauss2DsMLE, [409](#)
 - mappel::Gauss2DsModel, [441](#)
 - mappel::Gauss2DsxyMAP, [469](#), [470](#)
 - mappel::Gauss2DsxyModel, [495](#), [496](#)
 - mappel::MAPEstimator, [545](#)
 - mappel::MCMCAdaptor1Ds, [586](#)
 - mappel::MCMCAdaptor1D, [569](#)
 - mappel::MCMCAdaptor2Ds, [622](#)
 - mappel::MCMCAdaptor2D, [604](#)
 - mappel::MLEstimator, [642](#), [643](#)
 - mappel::PointEmitterModel, [688](#)
- make_prior_beta_position
 - mappel::Gauss1DMAP, [120](#)
 - mappel::Gauss1DMLE, [148](#)
 - mappel::Gauss1DModel, [176](#)
 - mappel::Gauss1DsMAP, [203](#)
 - mappel::Gauss1DsMLE, [231](#)

- 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::Gauss2DsxyModel, [496](#)
- make_prior_component_intensity
 - mappel::Gauss1DMAP, [120](#)
 - mappel::Gauss1DMLE, [148](#)
 - mappel::Gauss1DModel, [176](#)
 - mappel::Gauss1DsMAP, [203](#)
 - mappel::Gauss1DsMLE, [232](#)
 - mappel::Gauss1DsModel, [259](#)
 - mappel::Gauss2DMAP, [288](#)
 - mappel::Gauss2DMLE, [318](#)
 - mappel::Gauss2DModel, [347](#)
 - mappel::Gauss2DsMAP, [378](#)
 - mappel::Gauss2DsMLE, [410](#)
 - mappel::Gauss2DsModel, [441](#)
 - mappel::Gauss2DsxyMAP, [470](#)
 - mappel::Gauss2DsxyModel, [496](#)
 - mappel::MAPEstimator, [546](#)
 - mappel::MCMCAdaptor1Ds, [587](#)
 - mappel::MCMCAdaptor1D, [569](#)
 - mappel::MCMCAdaptor2Ds, [623](#)
 - mappel::MCMCAdaptor2D, [605](#)
 - mappel::MLEstimator, [643](#)
 - mappel::PointEmitterModel, [688](#)
- make_prior_component_position_beta
 - mappel::Gauss1DMAP, [121](#)
 - mappel::Gauss1DMLE, [149](#)
 - mappel::Gauss1DModel, [176](#)
 - mappel::Gauss1DsMAP, [204](#)
 - mappel::Gauss1DsMLE, [232](#)
 - mappel::Gauss1DsModel, [259](#)
 - mappel::Gauss2DMAP, [288](#)
 - mappel::Gauss2DMLE, [318](#)
 - mappel::Gauss2DModel, [347](#)
 - mappel::Gauss2DsMAP, [378](#)
 - mappel::Gauss2DsMLE, [410](#)
 - mappel::Gauss2DsModel, [442](#)
 - mappel::Gauss2DsxyMAP, [470](#)
 - mappel::Gauss2DsxyModel, [496](#)
 - mappel::MAPEstimator, [546](#)
 - mappel::MCMCAdaptor1Ds, [587](#)
 - mappel::MCMCAdaptor1D, [569](#)
 - mappel::MCMCAdaptor2Ds, [623](#)
 - mappel::MCMCAdaptor2D, [605](#)
 - mappel::MLEstimator, [643](#)
 - mappel::PointEmitterModel, [688](#)
- make_prior_component_position_normal
 - mappel::Gauss1DMAP, [121](#)
 - mappel::Gauss1DMLE, [149](#)
 - mappel::Gauss1DModel, [176](#)
 - mappel::Gauss1DsMAP, [204](#)
 - mappel::Gauss1DsMLE, [232](#)
 - mappel::Gauss1DsModel, [260](#)
 - mappel::Gauss2DMAP, [289](#)
 - mappel::Gauss2DMLE, [319](#)
 - mappel::Gauss2DModel, [348](#)
 - mappel::Gauss2DsMAP, [379](#)
- mappel::Gauss1DMAP, [121](#)
- mappel::Gauss1DMLE, [149](#)
- mappel::Gauss1DModel, [176](#)
- mappel::Gauss1DsMAP, [204](#)
- mappel::Gauss1DsMLE, [232](#)
- mappel::Gauss1DsModel, [260](#)
- mappel::Gauss2DMAP, [289](#)
- mappel::Gauss2DMLE, [319](#)
- mappel::Gauss2DModel, [348](#)
- mappel::Gauss2DsMAP, [378](#)
- mappel::Gauss2DsMLE, [410](#)
- mappel::Gauss2DsModel, [442](#)
- mappel::Gauss2DsxyMAP, [470](#)
- mappel::Gauss2DsxyModel, [496](#)
- mappel::MAPEstimator, [546](#)
- mappel::MCMCAdaptor1Ds, [587](#)
- mappel::MCMCAdaptor1D, [569](#)
- mappel::MCMCAdaptor2Ds, [623](#)
- mappel::MCMCAdaptor2D, [605](#)
- mappel::MLEstimator, [643](#)
- mappel::PointEmitterModel, [689](#)
- make_prior_component_sigma
 - mappel::Gauss1DMAP, [121](#)
 - mappel::Gauss1DMLE, [149](#)
 - mappel::Gauss1DModel, [176](#)
 - mappel::Gauss1DsMAP, [204](#)
 - mappel::Gauss1DsMLE, [232](#)
 - 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, [496](#)
 - mappel::MAPEstimator, [546](#)
 - mappel::MCMCAdaptor1Ds, [587](#)
 - mappel::MCMCAdaptor1D, [570](#)
 - mappel::MCMCAdaptor2Ds, [623](#)
 - mappel::MCMCAdaptor2D, [605](#)
 - mappel::MLEstimator, [643](#)
 - mappel::PointEmitterModel, [689](#)
- make_prior_normal_position
 - mappel::Gauss1DMAP, [121](#)
 - mappel::Gauss1DMLE, [149](#)
 - mappel::Gauss1DModel, [177](#)
 - mappel::Gauss1DsMAP, [204](#)
 - mappel::Gauss1DsMLE, [233](#)
 - mappel::Gauss1DsModel, [260](#)
 - mappel::Gauss2DMAP, [289](#)
 - mappel::Gauss2DMLE, [319](#)
 - mappel::Gauss2DModel, [348](#)
 - mappel::Gauss2DsMAP, [379](#)

- mappel::Gauss2DsMLE, 411
- mappel::Gauss2DsModel, 442
- mappel::Gauss2DsxyMAP, 471
- mappel::Gauss2DsxyModel, 497
- make_stencil
 - 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, 379
 - mappel::Gauss2DsMLE, 411
 - mappel::Gauss2DsModel, 442
 - mappel::Gauss2DsxyMAP, 471
 - mappel::Gauss2DsxyModel, 497
- 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](#)
- mappel::ArraySizeError, [86](#)
- ArraySizeError, [87](#)
- mappel::CGaussHeuristicEstimator
 - CGaussHeuristicEstimator, [88](#)
 - clear_stats, [88](#)
 - compute_estimate, [88](#)
 - compute_estimate_debug, [89](#)
 - compute_profile_estimate, [89](#)
 - estimate_max, [89](#), [90](#)
 - estimate_max_debug, [90](#)
 - estimate_max_stack, [90](#)
 - estimate_profile_stack, [90](#)
 - get_debug_stats, [91](#)
 - get_model, [91](#)
 - get_stats, [91](#)
 - max_threads, [92](#)
 - model, [92](#)
 - mtx, [92](#)
 - name, [91](#)
 - num_estimations, [92](#)
 - num_threads, [92](#)
 - record_walltime, [91](#)
 - set_model, [92](#)
 - total_walltime, [93](#)
- mappel::CGaussHeuristicEstimator< Model >, [87](#)
- mappel::CGaussMLE< Model >, [93](#)
- mappel::CGaussMLE
 - CGaussMLE, [95](#)
 - clear_stats, [95](#)
 - compute_estimate, [95](#)
 - compute_estimate_debug, [95](#)
 - compute_profile_estimate, [96](#)
 - estimate_max, [96](#), [97](#)
 - estimate_max_debug, [97](#)
 - estimate_max_stack, [97](#)
 - estimate_profile_stack, [97](#)
 - get_debug_stats, [97](#)
 - get_model, [98](#)
 - get_stats, [98](#)
 - max_iterations, [99](#)
 - max_threads, [99](#)
 - model, [99](#)
 - mtx, [99](#)
 - name, [98](#)
 - num_estimations, [99](#)
 - num_threads, [99](#)
 - record_walltime, [98](#)
 - set_model, [98](#)
 - total_walltime, [99](#)
- mappel::Estimator
 - ~Estimator, [101](#)
 - clear_stats, [102](#)
 - compute_estimate, [102](#)

compute_estimate_debug, 102
 compute_profile_estimate, 102
 estimate_max, 103
 estimate_max_debug, 103
 estimate_max_stack, 104
 estimate_profile_stack, 104
 Estimator, 101
 get_debug_stats, 104
 get_model, 104
 get_stats, 104
 model, 105
 name, 105
 num_estimations, 105
 operator<<, 105
 record_walltime, 105
 set_model, 105
 total_walltime, 106
 mappel::Estimator< Model >, 100
 mappel::Gauss1DMAP, 106
 bound_theta, 113
 bounded_theta, 113
 bounded_theta_stack, 113
 bounds_epsilon, 128
 check_image_shape, 113
 check_param_shape, 114
 check_psf_sigma, 114
 check_size, 114
 default_alpha_sigma, 128
 default_beta_pos, 128
 default_intensity_kappa, 128
 default_max_l, 128
 default_mean_l, 129
 default_pixel_mean_bg, 129
 default_sigma_pos, 129
 DefaultPriorType, 129
 DefaultSeperableInitEstimator, 129
 estimator_names, 129
 eta_bg, 130
 eta_l, 130
 eta_x, 130
 Gauss1DMAP, 112, 113
 get_hyperparam_index, 114
 get_hyperparam_names, 115
 get_hyperparam_value, 115
 get_hyperparams, 115
 get_image_from_stack, 115
 get_lbound, 115
 get_mcmc_num_phases, 115
 get_mcmc_sigma_scale, 115
 get_num_hyperparams, 116
 get_num_params, 116
 get_num_pixels, 116
 get_param_names, 116
 get_prior, 116
 get_psf_sigma, 116, 117
 get_rng_generator, 117
 get_rng_manager, 117
 get_size, 117
 get_size_image_stack, 117
 get_stats, 117
 get_ubound, 117
 global_default_mcmc_sigma_scale, 130
 global_max_mcmc_sigma_scale, 130
 global_max_psf_sigma, 130
 global_max_size, 131
 global_min_psf_sigma, 131
 global_min_size, 131
 has_hyperparam, 118
 ImageCoordT, 110
 ImagePixelT, 110
 ImageShapeT, 110
 ImageSizeShapeT, 110
 ImageSizeVecShapeT, 111
 ImageSizeVecT, 111
 ImageSizeT, 111
 ImageStackShapeT, 111
 ImageStackT, 111
 ImageT, 111
 initial_theta_estimate, 118
 lbound, 131
 make_default_prior, 118
 make_default_prior_beta_position, 118
 make_default_prior_normal_position, 118
 make_image, 119
 make_image_stack, 119
 make_param, 119
 make_param_mat, 119
 make_param_mat_stack, 120
 make_param_stack, 120
 make_prior_beta_position, 120
 make_prior_component_intensity, 120
 make_prior_component_position_beta, 121
 make_prior_component_position_normal, 121
 make_prior_component_sigma, 121
 make_prior_normal_position, 121
 make_stencil, 122
 ModelDataStackT, 111
 ModelDataT, 112
 name, 131
 num_dim, 131
 num_hyperparams, 132
 num_params, 132
 num_phases, 132
 operator=, 122
 ParamVecT, 112
 ParamT, 112
 pixel_grad, 123
 pixel_grad2, 123

- pixel_hess, 123
- pixel_hess_update, 123
- pixel_model_value, 123
- prior, 132
- prior_types, 132
- psf_sigma, 133
- reflected_theta, 123
- reflected_theta_stack, 124
- rename_hyperparam, 124
- sample_mcmc_candidate, 124
- sample_prior, 124
- set_background_mcmc_sampling, 124
- set_bounds, 125
- set_hyperparam_names, 125
- set_hyperparam_value, 125
- set_hyperparams, 125
- set_image_in_stack, 125
- set_intensity_mcmc_sampling, 125
- set_lbound, 126
- set_mcmc_num_phases, 126
- set_mcmc_sigma_scale, 126
- set_param_names, 126
- set_prior, 126
- set_psf_sigma, 127
- set_rng_seed, 127
- set_size, 127
- set_ubound, 127
- sigma_scale, 133
- size, 133
- StencilVecT, 112
- theta_in_bounds, 127
- theta_stack_in_bounds, 128
- ubound, 133
- mappel::Gauss1DMLE, 134
 - bound_theta, 141
 - bounded_theta, 141
 - bounded_theta_stack, 141
 - bounds_epsilon, 156
 - check_image_shape, 141
 - check_param_shape, 142
 - check_psf_sigma, 142
 - check_size, 142
 - default_alpha_sigma, 156
 - default_beta_pos, 156
 - default_intensity_kappa, 156
 - default_max_l, 156
 - default_mean_l, 157
 - default_pixel_mean_bg, 157
 - default_sigma_pos, 157
 - DefaultPriorType, 157
 - DefaultSeperableInitEstimator, 157
 - estimator_names, 157
 - eta_bg, 158
 - eta_l, 158
 - eta_x, 158
 - Gauss1DMLE, 140, 141
 - get_hyperparam_index, 142
 - get_hyperparam_names, 143
 - get_hyperparam_value, 143
 - get_hyperparams, 143
 - get_image_from_stack, 143
 - get_lbound, 143
 - get_mcmc_num_phases, 143
 - get_mcmc_sigma_scale, 143
 - get_num_hyperparams, 144
 - get_num_params, 144
 - get_num_pixels, 144
 - get_param_names, 144
 - get_prior, 144
 - get_psf_sigma, 144, 145
 - get_rng_generator, 145
 - get_rng_manager, 145
 - get_size, 145
 - get_size_image_stack, 145
 - get_stats, 145
 - get_ubound, 145
 - global_default_mcmc_sigma_scale, 158
 - global_max_mcmc_sigma_scale, 158
 - global_max_psf_sigma, 158
 - global_max_size, 159
 - global_min_psf_sigma, 159
 - global_min_size, 159
 - has_hyperparam, 146
 - ImageCoordT, 138
 - ImagePixelT, 138
 - ImageShapeT, 138
 - ImageSizeShapeT, 138
 - ImageSizeVecShapeT, 139
 - ImageSizeVecT, 139
 - ImageSizeT, 139
 - ImageStackShapeT, 139
 - ImageStackT, 139
 - ImageT, 139
 - initial_theta_estimate, 146
 - lbound, 159
 - make_default_prior, 146
 - make_default_prior_beta_position, 146
 - make_default_prior_normal_position, 146
 - make_image, 147
 - make_image_stack, 147
 - make_param, 147
 - make_param_mat, 147
 - make_param_mat_stack, 148
 - make_param_stack, 148
 - make_prior_beta_position, 148
 - make_prior_component_intensity, 148
 - make_prior_component_position_beta, 149
 - make_prior_component_position_normal, 149

make_prior_component_sigma, 149
 make_prior_normal_position, 149
 make_stencil, 150
 ModelDataStackT, 139
 ModelDataT, 140
 name, 159
 num_dim, 159
 num_hyperparams, 160
 num_params, 160
 num_phases, 160
 operator=, 150
 ParamVecT, 140
 ParamT, 140
 pixel_grad, 151
 pixel_grad2, 151
 pixel_hess, 151
 pixel_hess_update, 151
 pixel_model_value, 151
 prior, 160
 prior_types, 160
 psf_sigma, 161
 reflected_theta, 151
 reflected_theta_stack, 152
 rename_hyperparam, 152
 sample_mcmc_candidate, 152
 sample_prior, 152
 set_background_mcmc_sampling, 152
 set_bounds, 153
 set_hyperparam_names, 153
 set_hyperparam_value, 153
 set_hyperparams, 153
 set_image_in_stack, 153
 set_intensity_mcmc_sampling, 153
 set_lbound, 154
 set_mcmc_num_phases, 154
 set_mcmc_sigma_scale, 154
 set_param_names, 154
 set_prior, 154
 set_psf_sigma, 155
 set_rng_seed, 155
 set_size, 155
 set_ubound, 155
 sigma_scale, 161
 size, 161
 StencilVecT, 140
 theta_in_bounds, 155
 theta_stack_in_bounds, 156
 ubound, 161
 mappel::Gauss1DModel, 162
 bound_theta, 168
 bounded_theta, 168
 bounded_theta_stack, 169
 bounds_epsilon, 183
 check_image_shape, 169
 check_param_shape, 169
 check_psf_sigma, 169, 170
 check_size, 170
 default_alpha_sigma, 183
 default_beta_pos, 183
 default_intensity_kappa, 183
 default_max_l, 183
 default_mean_l, 184
 default_pixel_mean_bg, 184
 default_sigma_pos, 184
 DefaultPriorType, 184
 DefaultSeperableInitEstimator, 184
 eta_bg, 184
 eta_l, 185
 eta_x, 185
 Gauss1DModel, 168
 get_hyperparam_index, 170
 get_hyperparam_names, 170
 get_hyperparam_value, 170
 get_hyperparams, 170
 get_image_from_stack, 171
 get_lbound, 171
 get_mcmc_num_phases, 171
 get_mcmc_sigma_scale, 171
 get_num_hyperparams, 171
 get_num_params, 171
 get_num_pixels, 171
 get_param_names, 172
 get_prior, 172
 get_psf_sigma, 172
 get_rng_generator, 172
 get_rng_manager, 172
 get_size, 173
 get_size_image_stack, 173
 get_stats, 173
 get_ubound, 173
 global_default_mcmc_sigma_scale, 185
 global_max_mcmc_sigma_scale, 185
 global_max_psf_sigma, 185
 global_max_size, 186
 global_min_psf_sigma, 186
 global_min_size, 186
 has_hyperparam, 173
 ImageCoordT, 166
 ImagePixelT, 166
 ImageShapeT, 166
 ImageSizeShapeT, 167
 ImageSizeVecShapeT, 167
 ImageSizeVecT, 167
 ImageSizeT, 167
 ImageStackShapeT, 167
 ImageStackT, 167
 ImageT, 167
 initial_theta_estimate, 173, 174

- lbound, 186
- make_default_prior, 174
- make_default_prior_beta_position, 174
- make_default_prior_normal_position, 174
- make_image, 174
- make_image_stack, 174
- make_param, 174, 175
- make_param_mat, 175
- make_param_mat_stack, 175
- make_param_stack, 175
- make_prior_beta_position, 176
- make_prior_component_intensity, 176
- make_prior_component_position_beta, 176
- make_prior_component_position_normal, 176
- make_prior_component_sigma, 176
- make_prior_normal_position, 177
- make_stencil, 177
- num_dim, 186
- num_hyperparams, 187
- num_params, 187
- num_phases, 187
- operator=, 177, 178
- ParamVecT, 168
- ParamT, 168
- pixel_grad, 178
- pixel_grad2, 178
- pixel_hess, 178
- pixel_hess_update, 178
- pixel_model_value, 178
- prior, 187
- prior_types, 187
- psf_sigma, 188
- reflected_theta, 178
- reflected_theta_stack, 179
- rename_hyperparam, 179
- sample_mcmc_candidate, 179
- sample_prior, 179
- set_background_mcmc_sampling, 179
- set_bounds, 180
- set_hyperparam_names, 180
- set_hyperparam_value, 180
- set_hyperparams, 180
- set_image_in_stack, 180
- set_intensity_mcmc_sampling, 180
- set_lbound, 181
- set_mcmc_num_phases, 181
- set_mcmc_sigma_scale, 181
- set_param_names, 181
- set_prior, 181
- set_psf_sigma, 182
- set_rng_seed, 182
- set_size, 182
- set_ubound, 182
- sigma_scale, 188
- size, 188
- StencilVecT, 168
- theta_in_bounds, 182
- theta_stack_in_bounds, 183
- ubound, 188
- mappel::Gauss1DModel::Stencil, 734
 - bg, 736
 - compute_derivatives, 736
 - DXS, 737
 - derivatives_computed, 737
 - DX, 737
 - dx, 737
 - Gx, 737
 - I, 736
 - model, 738
 - operator<<, 737
 - ParamT, 735
 - Stencil, 736
 - theta, 738
 - X, 738
 - x, 736
- mappel::Gauss1DsMAP, 189
 - bound_theta, 196
 - bounded_theta, 196
 - bounded_theta_stack, 196
 - bounds_epsilon, 211
 - check_image_shape, 196
 - check_param_shape, 197
 - check_psf_sigma, 197
 - check_size, 197
 - default_alpha_sigma, 211
 - default_beta_pos, 211
 - default_intensity_kappa, 212
 - default_max_I, 212
 - default_mean_I, 212
 - default_pixel_mean_bg, 212
 - default_sigma_pos, 212
 - DefaultPriorType, 212
 - DefaultSeperableInitEstimator, 213
 - estimator_names, 213
 - eta_bg, 213
 - eta_I, 213
 - eta_sigma, 213
 - eta_x, 213
 - Gauss1DsMAP, 195, 196
 - get_hyperparam_index, 197
 - get_hyperparam_names, 198
 - get_hyperparam_value, 198
 - get_hyperparams, 198
 - get_image_from_stack, 198
 - get_lbound, 198
 - get_max_sigma, 198
 - get_mcmc_num_phases, 198
 - get_mcmc_sigma_scale, 199

get_min_sigma, 199
get_num_hyperparams, 199
get_num_params, 199
get_num_pixels, 199
get_param_names, 199
get_prior, 199, 200
get_rng_generator, 200
get_rng_manager, 200
get_size, 200
get_size_image_stack, 200
get_stats, 200
get_ubound, 201
global_default_mcmc_sigma_scale, 214
global_max_mcmc_sigma_scale, 214
global_max_psf_sigma, 214
global_max_size, 214
global_min_psf_sigma, 214
global_min_size, 214
has_hyperparam, 201
ImageCoordT, 193
ImagePixelT, 193
ImageShapeT, 193
ImageSizeShapeT, 193
ImageSizeVecShapeT, 194
ImageSizeVecT, 194
ImageSizeT, 194
ImageStackShapeT, 194
ImageStackT, 194
ImageT, 194
initial_theta_estimate, 201
lbound, 215
make_default_prior, 201
make_default_prior_beta_position, 201
make_default_prior_normal_position, 202
make_image, 202
make_image_stack, 202
make_param, 202
make_param_mat, 202, 203
make_param_mat_stack, 203
make_param_stack, 203
make_prior_beta_position, 203
make_prior_component_intensity, 203
make_prior_component_position_beta, 204
make_prior_component_position_normal, 204
make_prior_component_sigma, 204
make_prior_normal_position, 204
make_stencil, 205
ModelDataStackT, 194
ModelDataT, 195
name, 215
num_dim, 215
num_hyperparams, 215
num_params, 215
num_phases, 215
operator=, 205
ParamVecT, 195
ParamT, 195
pixel_grad, 206
pixel_grad2, 206
pixel_hess, 206
pixel_hess_update, 206
pixel_model_value, 206
prior, 216
prior_types, 216
reflected_theta, 206
reflected_theta_stack, 207
rename_hyperparam, 207
sample_mcmc_candidate, 207
sample_prior, 207
set_background_mcmc_sampling, 207
set_bounds, 208
set_hyperparam_names, 208
set_hyperparam_value, 208
set_hyperparams, 208
set_image_in_stack, 208
set_intensity_mcmc_sampling, 208
set_lbound, 209
set_max_sigma, 209
set_mcmc_num_phases, 209
set_mcmc_sigma_scale, 209
set_min_sigma, 209, 210
set_param_names, 210
set_prior, 210
set_rng_seed, 210
set_size, 210
set_ubound, 211
sigma_scale, 216
size, 216
StencilVecT, 195
theta_in_bounds, 211
theta_stack_in_bounds, 211
ubound, 217
mappel::Gauss1DsMLE, 217
bound_theta, 224
bounded_theta, 224
bounded_theta_stack, 224
bounds_epsilon, 239
check_image_shape, 225
check_param_shape, 225
check_psf_sigma, 225
check_size, 226
default_alpha_sigma, 239
default_beta_pos, 240
default_intensity_kappa, 240
default_max_l, 240
default_mean_l, 240
default_pixel_mean_bg, 240
default_sigma_pos, 240

DefaultPriorType, 241
DefaultSeperableInitEstimator, 241
estimator_names, 241
eta_bg, 241
eta_l, 241
eta_sigma, 241
eta_x, 242
Gauss1DsMLE, 224
get_hyperparam_index, 226
get_hyperparam_names, 226
get_hyperparam_value, 226
get_hyperparams, 226
get_image_from_stack, 226
get_lbound, 226
get_max_sigma, 227
get_mcmc_num_phases, 227
get_mcmc_sigma_scale, 227
get_min_sigma, 227
get_num_hyperparams, 227
get_num_params, 227
get_num_pixels, 227
get_param_names, 228
get_prior, 228
get_rng_generator, 228
get_rng_manager, 228
get_size, 228
get_size_image_stack, 229
get_stats, 229
get_ubound, 229
global_default_mcmc_sigma_scale, 242
global_max_mcmc_sigma_scale, 242
global_max_psf_sigma, 242
global_max_size, 242
global_min_psf_sigma, 242
global_min_size, 243
has_hyperparam, 229
ImageCoordT, 222
ImagePixelT, 222
ImageShapeT, 222
ImageSizeShapeT, 222
ImageSizeVecShapeT, 222
ImageSizeVecT, 222
ImageSizeT, 222
ImageStackShapeT, 222
ImageStackT, 223
ImageT, 223
initial_theta_estimate, 229
lbound, 243
make_default_prior, 229
make_default_prior_beta_position, 230
make_default_prior_normal_position, 230
make_image, 230
make_image_stack, 230
make_param, 230
make_param_mat, 231
make_param_mat_stack, 231
make_param_stack, 231
make_prior_beta_position, 231
make_prior_component_intensity, 232
make_prior_component_position_beta, 232
make_prior_component_position_normal, 232
make_prior_component_sigma, 232
make_prior_normal_position, 233
make_stencil, 233
ModelDataStackT, 223
ModelDataT, 223
name, 243
num_dim, 243
num_hyperparams, 243
num_params, 243
num_phases, 244
operator=, 233, 234
ParamVecT, 223
ParamT, 223
pixel_grad, 234
pixel_grad2, 234
pixel_hess, 234
pixel_hess_update, 234
pixel_model_value, 234
prior, 244
prior_types, 244
reflected_theta, 235
reflected_theta_stack, 235
rename_hyperparam, 235
sample_mcmc_candidate, 235
sample_prior, 235
set_background_mcmc_sampling, 236
set_bounds, 236
set_hyperparam_names, 236
set_hyperparam_value, 236
set_hyperparams, 236
set_image_in_stack, 236
set_intensity_mcmc_sampling, 237
set_lbound, 237
set_max_sigma, 237
set_mcmc_num_phases, 237
set_mcmc_sigma_scale, 237
set_min_sigma, 238
set_param_names, 238
set_prior, 238
set_rng_seed, 238
set_size, 238, 239
set_ubound, 239
sigma_scale, 244
size, 245
StencilVecT, 223
theta_in_bounds, 239
theta_stack_in_bounds, 239

- ubound, 245
- mappel::Gauss1DsModel, 245
 - bound_theta, 252
 - bounded_theta, 252
 - bounded_theta_stack, 252
 - bounds_epsilon, 267
 - check_image_shape, 252
 - check_param_shape, 252
 - check_psf_sigma, 253
 - check_size, 253
 - default_alpha_sigma, 267
 - default_beta_pos, 267
 - default_intensity_kappa, 267
 - default_max_l, 267
 - default_mean_l, 267
 - default_pixel_mean_bg, 267
 - default_sigma_pos, 268
 - DefaultPriorType, 268
 - DefaultSeperableInitEstimator, 268
 - eta_bg, 268
 - eta_l, 268
 - eta_sigma, 268
 - eta_x, 269
 - Gauss1DsModel, 251
 - get_hyperparam_index, 253
 - get_hyperparam_names, 253
 - get_hyperparam_value, 253
 - get_hyperparams, 254
 - get_image_from_stack, 254
 - get_lbound, 254
 - get_max_sigma, 254
 - get_mcmc_num_phases, 254
 - get_mcmc_sigma_scale, 254
 - get_min_sigma, 254
 - get_num_hyperparams, 255
 - get_num_params, 255
 - get_num_pixels, 255
 - get_param_names, 255
 - get_prior, 255
 - get_rng_generator, 255
 - get_rng_manager, 256
 - get_size, 256
 - get_size_image_stack, 256
 - get_stats, 256
 - get_ubound, 256
 - global_default_mcmc_sigma_scale, 269
 - global_max_mcmc_sigma_scale, 269
 - global_max_psf_sigma, 269
 - global_max_size, 269
 - global_min_psf_sigma, 269
 - global_min_size, 270
 - has_hyperparam, 256
 - ImageCoordT, 249
 - ImagePixelT, 249
 - ImageShapeT, 250
 - ImageSizeShapeT, 250
 - ImageSizeVecShapeT, 250
 - ImageSizeVecT, 250
 - ImageSizeT, 250
 - ImageStackShapeT, 250
 - ImageStackT, 250
 - ImageT, 251
 - initial_theta_estimate, 257
 - lbound, 270
 - make_default_prior, 257
 - make_default_prior_beta_position, 257
 - make_default_prior_normal_position, 257
 - make_image, 257
 - make_image_stack, 258
 - make_param, 258
 - make_param_mat, 258
 - make_param_mat_stack, 258
 - make_param_stack, 259
 - make_prior_beta_position, 259
 - make_prior_component_intensity, 259
 - make_prior_component_position_beta, 259
 - make_prior_component_position_normal, 260
 - make_prior_component_sigma, 260
 - make_prior_normal_position, 260
 - make_stencil, 260
 - num_dim, 270
 - num_hyperparams, 270
 - num_params, 270
 - num_phases, 270
 - operator=, 261
 - ParamVecT, 251
 - ParamT, 251
 - pixel_grad, 261
 - pixel_grad2, 261
 - pixel_hess, 261
 - pixel_hess_update, 262
 - pixel_model_value, 262
 - prior, 271
 - prior_types, 271
 - reflected_theta, 262
 - reflected_theta_stack, 262
 - rename_hyperparam, 262
 - sample_mcmc_candidate, 262
 - sample_prior, 263
 - set_background_mcmc_sampling, 263
 - set_bounds, 263
 - set_hyperparam_names, 263
 - set_hyperparam_value, 263
 - set_hyperparams, 264
 - set_image_in_stack, 264
 - set_intensity_mcmc_sampling, 264
 - set_lbound, 264
 - set_max_sigma, 264

- set_mcmc_num_phases, 265
- set_mcmc_sigma_scale, 265
- set_min_sigma, 265
- set_param_names, 265
- set_prior, 265
- set_rng_seed, 266
- set_size, 266
- set_ubound, 266
- sigma_scale, 271
- size, 271
- StencilVecT, 251
- theta_in_bounds, 266
- theta_stack_in_bounds, 266
- ubound, 272
- mappel::Gauss1DsModel::Stencil, 738
 - bg, 740
 - compute_derivatives, 740
 - DXS2, 741
 - DXSX, 741
 - DXS, 741
 - derivatives_computed, 741
 - DX, 741
 - dx, 741
 - Gx, 741
 - I, 740
 - model, 742
 - operator<<, 741
 - ParamT, 739
 - sigma, 740
 - Stencil, 739
 - theta, 742
 - X, 742
 - x, 740
- mappel::Gauss2DMAP, 272
 - bound_theta, 280
 - bounded_theta, 280
 - bounded_theta_stack, 280
 - bounds_epsilon, 295
 - check_image_shape, 280
 - check_param_shape, 280, 281
 - check_psf_sigma, 281
 - check_size, 281
 - debug_internal_sum_model_x, 281
 - debug_internal_sum_model_y, 281
 - default_alpha_sigma, 295
 - default_beta_pos, 295
 - default_intensity_kappa, 296
 - default_max_I, 296
 - default_mean_I, 296
 - default_pixel_mean_bg, 296
 - default_sigma_pos, 296
 - DefaultPriorType, 296
 - DefaultSeperableInitEstimator, 297
 - estimator_names, 297
 - eta_bg, 297
 - eta_I, 297
 - eta_x, 297
 - eta_y, 297
 - Gauss1DSumModelT, 277
 - Gauss2DMAP, 279, 280
 - get_hyperparam_index, 282
 - get_hyperparam_names, 282
 - get_hyperparam_value, 282
 - get_hyperparams, 282
 - get_image_from_stack, 282
 - get_lbound, 282
 - get_mcmc_num_phases, 282
 - get_mcmc_sigma_scale, 283
 - get_num_hyperparams, 283
 - get_num_params, 283
 - get_num_pixels, 283
 - get_param_names, 283
 - get_prior, 283
 - get_psf_sigma, 284
 - get_rng_generator, 284
 - get_rng_manager, 284
 - get_size, 284
 - get_size_image_stack, 284
 - get_stats, 285
 - get_ubound, 285
 - global_default_mcmc_sigma_scale, 298
 - global_max_mcmc_sigma_scale, 298
 - global_max_psf_sigma, 298
 - global_max_size, 298
 - global_min_psf_sigma, 298
 - global_min_size, 298
 - has_hyperparam, 285
 - ImageCoordT, 277
 - ImagePixelT, 277
 - ImageShapeT, 277
 - ImageSizeShapeT, 277
 - ImageSizeVecShapeT, 277
 - ImageSizeVecT, 278
 - ImageSizeT, 277
 - ImageStackShapeT, 278
 - ImageStackT, 278
 - ImageT, 278
 - initial_theta_estimate, 285
 - lbound, 299
 - make_default_prior, 286
 - make_default_prior_beta_position, 286
 - make_default_prior_normal_position, 286
 - make_image, 286
 - make_image_stack, 286
 - make_internal_1Dsum_estimator, 286
 - make_param, 287
 - make_param_mat, 287
 - make_param_mat_stack, 287

[make_param_stack](#), 288
[make_prior_beta_position](#), 288
[make_prior_component_intensity](#), 288
[make_prior_component_position_beta](#), 288
[make_prior_component_position_normal](#), 289
[make_prior_component_sigma](#), 289
[make_prior_normal_position](#), 289
[make_stencil](#), 289
[ModelDataStackT](#), 278
[ModelDataT](#), 278
[name](#), 299
[num_dim](#), 299
[num_hyperparams](#), 299
[num_params](#), 299
[num_phases](#), 299
[operator=](#), 290
[ParamVecT](#), 279
[ParamT](#), 278
[pixel_grad](#), 290
[pixel_grad2](#), 290
[pixel_hess](#), 290
[pixel_hess_update](#), 291
[pixel_model_value](#), 291
[prior](#), 300
[prior_types](#), 300
[psf_sigma](#), 300
[reflected_theta](#), 291
[reflected_theta_stack](#), 291
[rename_hyperparam](#), 291
[sample_mcmc_candidate](#), 291
[sample_prior](#), 292
[set_background_mcmc_sampling](#), 292
[set_bounds](#), 292
[set_hyperparam_names](#), 292
[set_hyperparam_value](#), 292
[set_hyperparams](#), 293
[set_image_in_stack](#), 293
[set_intensity_mcmc_sampling](#), 293
[set_lbound](#), 293
[set_mcmc_num_phases](#), 293
[set_mcmc_sigma_scale](#), 293
[set_param_names](#), 293
[set_prior](#), 294
[set_psf_sigma](#), 294
[set_rng_seed](#), 294
[set_size](#), 294
[set_ubound](#), 294
[sigma_scale](#), 300
[size](#), 301
[StencilVecT](#), 279
[theta_in_bounds](#), 294
[theta_stack_in_bounds](#), 295
[ubound](#), 301
[update_internal_1Dsum_estimators](#), 295
[x_model](#), 301
[y_model](#), 302
[mappel::Gauss2DMLE](#), 302
[bound_theta](#), 310
[bounded_theta](#), 310
[bounded_theta_stack](#), 310
[bounds_epsilon](#), 325
[check_image_shape](#), 310
[check_param_shape](#), 310, 311
[check_psf_sigma](#), 311
[check_size](#), 311
[debug_internal_sum_model_x](#), 311
[debug_internal_sum_model_y](#), 311
[default_alpha_sigma](#), 325
[default_beta_pos](#), 325
[default_intensity_kappa](#), 326
[default_max_l](#), 326
[default_mean_l](#), 326
[default_pixel_mean_bg](#), 326
[default_sigma_pos](#), 326
[DefaultPriorType](#), 326
[DefaultSeperableInitEstimator](#), 327
[estimator_names](#), 327
[eta_bg](#), 327
[eta_l](#), 327
[eta_x](#), 327
[eta_y](#), 327
[Gauss1DSumModelT](#), 307
[Gauss2DMLE](#), 309
[get_hyperparam_index](#), 312
[get_hyperparam_names](#), 312
[get_hyperparam_value](#), 312
[get_hyperparams](#), 312
[get_image_from_stack](#), 312
[get_lbound](#), 312
[get_mcmc_num_phases](#), 312
[get_mcmc_sigma_scale](#), 313
[get_num_hyperparams](#), 313
[get_num_params](#), 313
[get_num_pixels](#), 313
[get_param_names](#), 313
[get_prior](#), 313
[get_psf_sigma](#), 314
[get_rng_generator](#), 314
[get_rng_manager](#), 314
[get_size](#), 314
[get_size_image_stack](#), 314
[get_stats](#), 315
[get_ubound](#), 315
[global_default_mcmc_sigma_scale](#), 328
[global_max_mcmc_sigma_scale](#), 328
[global_max_psf_sigma](#), 328
[global_max_size](#), 328
[global_min_psf_sigma](#), 328

global_min_size, 328
has_hyperparam, 315
ImageCoordT, 307
ImagePixelT, 307
ImageShapeT, 307
ImageSizeShapeT, 307
ImageSizeVecShapeT, 307
ImageSizeVecT, 308
ImageSizeT, 307
ImageStackShapeT, 308
ImageStackT, 308
ImageT, 308
initial_theta_estimate, 315
lbound, 329
make_default_prior, 316
make_default_prior_beta_position, 316
make_default_prior_normal_position, 316
make_image, 316
make_image_stack, 316
make_internal_1Dsum_estimator, 316
make_param, 317
make_param_mat, 317
make_param_mat_stack, 317
make_param_stack, 318
make_prior_beta_position, 318
make_prior_component_intensity, 318
make_prior_component_position_beta, 318
make_prior_component_position_normal, 319
make_prior_component_sigma, 319
make_prior_normal_position, 319
make_stencil, 319
ModelDataStackT, 308
ModelDataT, 308
name, 329
num_dim, 329
num_hyperparams, 329
num_params, 329
num_phases, 329
operator=, 320
ParamVecT, 309
ParamT, 308
pixel_grad, 320
pixel_grad2, 320
pixel_hess, 320
pixel_hess_update, 321
pixel_model_value, 321
prior, 330
prior_types, 330
psf_sigma, 330
reflected_theta, 321
reflected_theta_stack, 321
rename_hyperparam, 321
sample_mcmc_candidate, 321
sample_prior, 322
set_background_mcmc_sampling, 322
set_bounds, 322
set_hyperparam_names, 322
set_hyperparam_value, 322
set_hyperparams, 323
set_image_in_stack, 323
set_intensity_mcmc_sampling, 323
set_lbound, 323
set_mcmc_num_phases, 323
set_mcmc_sigma_scale, 323
set_param_names, 323
set_prior, 324
set_psf_sigma, 324
set_rng_seed, 324
set_size, 324
set_ubound, 324
sigma_scale, 330
size, 331
StencilVecT, 309
theta_in_bounds, 324
theta_stack_in_bounds, 325
ubound, 331
update_internal_1Dsum_estimators, 325
x_model, 331
y_model, 332
mappel::Gauss2DModel, 332
bound_theta, 339
bounded_theta, 339
bounded_theta_stack, 339
bounds_epsilon, 354
check_image_shape, 339
check_param_shape, 340
check_psf_sigma, 340
check_size, 340
debug_internal_sum_model_x, 340
debug_internal_sum_model_y, 341
default_alpha_sigma, 354
default_beta_pos, 354
default_intensity_kappa, 354
default_max_l, 355
default_mean_l, 355
default_pixel_mean_bg, 355
default_sigma_pos, 355
DefaultPriorType, 355
DefaultSeperableInitEstimator, 355
eta_bg, 356
eta_l, 356
eta_x, 356
eta_y, 356
Gauss1DSumModelT, 336
Gauss2DModel, 338, 339
get_hyperparam_index, 341
get_hyperparam_names, 341
get_hyperparam_value, 341

get_hyperparams, 341
get_image_from_stack, 341
get_lbound, 341
get_mcmc_num_phases, 342
get_mcmc_sigma_scale, 342
get_num_hyperparams, 342
get_num_params, 342
get_num_pixels, 342
get_param_names, 342
get_prior, 342, 343
get_psf_sigma, 343
get_rng_generator, 343
get_rng_manager, 343
get_size, 343
get_size_image_stack, 344
get_stats, 344
get_ubound, 344
global_default_mcmc_sigma_scale, 356
global_max_mcmc_sigma_scale, 357
global_max_psf_sigma, 357
global_max_size, 357
global_min_psf_sigma, 357
global_min_size, 357
has_hyperparam, 344
ImageCoordT, 336
ImagePixelT, 337
ImageShapeT, 337
ImageSizeShapeT, 337
ImageSizeVecShapeT, 337
ImageSizeVecT, 337
ImageSizeT, 337
ImageStackShapeT, 337
ImageStackT, 338
ImageT, 338
initial_theta_estimate, 344, 345
lbound, 357
make_default_prior, 345
make_default_prior_beta_position, 345
make_default_prior_normal_position, 345
make_image, 345
make_image_stack, 345
make_internal_1Dsum_estimator, 346
make_param, 346
make_param_mat, 346
make_param_mat_stack, 346, 347
make_param_stack, 347
make_prior_beta_position, 347
make_prior_component_intensity, 347
make_prior_component_position_beta, 347
make_prior_component_position_normal, 348
make_prior_component_sigma, 348
make_prior_normal_position, 348
make_stencil, 348
num_dim, 358
num_hyperparams, 358
num_params, 358
num_phases, 358
operator=, 349
ParamVecT, 338
ParamT, 338
pixel_grad, 349
pixel_grad2, 349
pixel_hess, 349
pixel_hess_update, 350
pixel_model_value, 350
prior, 358
prior_types, 359
psf_sigma, 359
reflected_theta, 350
reflected_theta_stack, 350
rename_hyperparam, 350
sample_mcmc_candidate, 350
sample_prior, 351
set_background_mcmc_sampling, 351
set_bounds, 351
set_hyperparam_names, 351
set_hyperparam_value, 351
set_hyperparams, 352
set_image_in_stack, 352
set_intensity_mcmc_sampling, 352
set_lbound, 352
set_mcmc_num_phases, 352
set_mcmc_sigma_scale, 352
set_param_names, 352
set_prior, 353
set_psf_sigma, 353
set_rng_seed, 353
set_size, 353
set_ubound, 353
sigma_scale, 359
size, 359
StencilVecT, 338
theta_in_bounds, 353
theta_stack_in_bounds, 354
ubound, 360
update_internal_1Dsum_estimators, 354
x_model, 360
y_model, 360
mappel::Gauss2DModel::Stencil, 753
bg, 755
compute_derivatives, 755
DXS, 756
DYS, 757
derivatives_computed, 756
DX, 756
dx, 756
DY, 757
dy, 756

- Gx, [757](#)
- Gy, [757](#)
- I, [755](#)
- model, [757](#)
- operator<<, [756](#)
- ParamT, [754](#)
- Stencil, [754](#)
- theta, [757](#)
- X, [757](#)
- x, [755](#)
- Y, [757](#)
- y, [755](#)
- mappel::Gauss2DsMAP, [361](#)
 - bound_theta, [368](#)
 - bounded_theta, [368](#)
 - bounded_theta_stack, [369](#)
 - bounds_epsilon, [386](#)
 - check_image_shape, [369](#)
 - check_param_shape, [369](#)
 - check_psf_sigma, [369](#), [370](#)
 - check_size, [370](#)
 - compute_max_sigma_ratio, [370](#)
 - debug_internal_sum_model_x, [370](#)
 - debug_internal_sum_model_y, [370](#)
 - default_alpha_sigma, [386](#)
 - default_beta_pos, [386](#)
 - default_intensity_kappa, [386](#)
 - default_max_l, [386](#)
 - default_mean_l, [386](#)
 - default_pixel_mean_bg, [386](#)
 - default_sigma_pos, [387](#)
 - DefaultPriorType, [387](#)
 - DefaultSeperableInitEstimator, [387](#)
 - estimator_names, [387](#)
 - eta_bg, [387](#)
 - eta_l, [387](#)
 - eta_sigma, [388](#)
 - eta_x, [388](#)
 - eta_y, [388](#)
 - Gauss1DSumModelT, [365](#)
 - Gauss2DsMAP, [368](#)
 - get_hyperparam_index, [371](#)
 - get_hyperparam_names, [371](#)
 - get_hyperparam_value, [371](#)
 - get_hyperparams, [371](#)
 - get_image_from_stack, [371](#)
 - get_lbound, [371](#)
 - get_max_sigma, [371](#), [372](#)
 - get_max_sigma_ratio, [372](#)
 - get_mcmc_num_phases, [372](#)
 - get_mcmc_sigma_scale, [372](#)
 - get_min_sigma, [372](#)
 - get_num_hyperparams, [372](#)
 - get_num_params, [373](#)
 - get_num_pixels, [373](#)
 - get_param_names, [373](#)
 - get_prior, [373](#)
 - get_rng_generator, [373](#)
 - get_rng_manager, [373](#)
 - get_size, [374](#)
 - get_size_image_stack, [374](#)
 - get_stats, [374](#)
 - get_ubound, [374](#)
 - global_default_mcmc_sigma_scale, [388](#)
 - global_max_mcmc_sigma_scale, [388](#)
 - global_max_psf_sigma, [389](#)
 - global_max_size, [389](#)
 - global_min_psf_sigma, [389](#)
 - global_min_size, [389](#)
 - has_hyperparam, [374](#)
 - ImageCoordT, [365](#)
 - ImagePixelT, [366](#)
 - ImageShapeT, [366](#)
 - ImageSizeShapeT, [366](#)
 - ImageSizeVecShapeT, [366](#)
 - ImageSizeVecT, [366](#)
 - ImageSizeT, [366](#)
 - ImageStackShapeT, [366](#)
 - ImageStackT, [367](#)
 - ImageT, [367](#)
 - initial_theta_estimate, [374](#), [375](#)
 - lbound, [389](#)
 - make_default_prior, [375](#)
 - make_default_prior_beta_position, [375](#)
 - make_default_prior_normal_position, [375](#)
 - make_image, [376](#)
 - make_image_stack, [376](#)
 - make_internal_1Dsum_estimator, [376](#)
 - make_param, [376](#)
 - make_param_mat, [377](#)
 - make_param_mat_stack, [377](#)
 - make_param_stack, [377](#)
 - make_prior_beta_position, [377](#)
 - make_prior_component_intensity, [378](#)
 - make_prior_component_position_beta, [378](#)
 - make_prior_component_position_normal, [378](#)
 - make_prior_component_sigma, [378](#)
 - make_prior_normal_position, [379](#)
 - make_stencil, [379](#)
 - min_sigma, [389](#)
 - ModelDataStackT, [367](#)
 - ModelDataT, [367](#)
 - name, [390](#)
 - num_dim, [390](#)
 - num_hyperparams, [390](#)
 - num_params, [390](#)
 - num_phases, [390](#)
 - operator=, [379](#), [380](#)

- ParamVecT, 367
- ParamT, 367
- pixel_grad, 380
- pixel_grad2, 380
- pixel_hess, 380
- pixel_hess_update, 380
- pixel_model_value, 381
- prior, 391
- prior_types, 391
- reflected_theta, 381
- reflected_theta_stack, 381
- rename_hyperparam, 381
- sample_mcmc_candidate, 381
- sample_prior, 381, 382
- set_background_mcmc_sampling, 382
- set_bounds, 382
- set_hyperparam_names, 382
- set_hyperparam_value, 382
- set_hyperparams, 382
- set_image_in_stack, 383
- set_intensity_mcmc_sampling, 383
- set_lbound, 383
- set_max_sigma, 383
- set_max_sigma_ratio, 383
- set_mcmc_num_phases, 383
- set_mcmc_sigma_scale, 384
- set_min_sigma, 384
- set_param_names, 384
- set_prior, 384
- set_rng_seed, 384
- set_size, 384
- set_ubound, 385
- sigma_scale, 391
- size, 391
- StencilVecT, 367
- theta_in_bounds, 385
- theta_stack_in_bounds, 385
- ubound, 392
- update_internal_1Dsum_estimators, 385
- x_model, 392
- y_model, 392
- mappel::Gauss2DsMLE, 393
 - bound_theta, 400
 - bounded_theta, 400
 - bounded_theta_stack, 401
 - bounds_epsilon, 418
 - check_image_shape, 401
 - check_param_shape, 401
 - check_psf_sigma, 401, 402
 - check_size, 402
 - compute_max_sigma_ratio, 402
 - debug_internal_sum_model_x, 402
 - debug_internal_sum_model_y, 402
 - default_alpha_sigma, 418
 - default_beta_pos, 418
 - default_intensity_kappa, 418
 - default_max_l, 418
 - default_mean_l, 418
 - default_pixel_mean_bg, 418
 - default_sigma_pos, 419
 - DefaultPriorType, 419
 - DefaultSeperableInitEstimator, 419
 - estimator_names, 419
 - eta_bg, 419
 - eta_l, 419
 - eta_sigma, 420
 - eta_x, 420
 - eta_y, 420
 - Gauss1DSumModelT, 397
 - Gauss2DsMLE, 400
 - get_hyperparam_index, 403
 - get_hyperparam_names, 403
 - get_hyperparam_value, 403
 - get_hyperparams, 403
 - get_image_from_stack, 403
 - get_lbound, 403
 - get_max_sigma, 403, 404
 - get_max_sigma_ratio, 404
 - get_mcmc_num_phases, 404
 - get_mcmc_sigma_scale, 404
 - get_min_sigma, 404
 - get_num_hyperparams, 404
 - get_num_params, 405
 - get_num_pixels, 405
 - get_param_names, 405
 - get_prior, 405
 - get_rng_generator, 405
 - get_rng_manager, 405
 - get_size, 406
 - get_size_image_stack, 406
 - get_stats, 406
 - get_ubound, 406
 - global_default_mcmc_sigma_scale, 420
 - global_max_mcmc_sigma_scale, 420
 - global_max_psf_sigma, 421
 - global_max_size, 421
 - global_min_psf_sigma, 421
 - global_min_size, 421
 - has_hyperparam, 406
 - ImageCoordT, 397
 - ImagePixelT, 398
 - ImageShapeT, 398
 - ImageSizeShapeT, 398
 - ImageSizeVecShapeT, 398
 - ImageSizeVecT, 398
 - ImageSizeT, 398
 - ImageStackShapeT, 398
 - ImageStackT, 399

ImageT, 399
initial_theta_estimate, 406, 407
lbound, 421
make_default_prior, 407
make_default_prior_beta_position, 407
make_default_prior_normal_position, 407
make_image, 408
make_image_stack, 408
make_internal_1Dsum_estimator, 408
make_param, 408
make_param_mat, 409
make_param_mat_stack, 409
make_param_stack, 409
make_prior_beta_position, 409
make_prior_component_intensity, 410
make_prior_component_position_beta, 410
make_prior_component_position_normal, 410
make_prior_component_sigma, 410
make_prior_normal_position, 411
make_stencil, 411
min_sigma, 421
ModelDataStackT, 399
ModelDataT, 399
name, 422
num_dim, 422
num_hyperparams, 422
num_params, 422
num_phases, 422
operator=, 411, 412
ParamVecT, 399
ParamT, 399
pixel_grad, 412
pixel_grad2, 412
pixel_hess, 412
pixel_hess_update, 412
pixel_model_value, 413
prior, 423
prior_types, 423
reflected_theta, 413
reflected_theta_stack, 413
rename_hyperparam, 413
sample_mcmc_candidate, 413
sample_prior, 413, 414
set_background_mcmc_sampling, 414
set_bounds, 414
set_hyperparam_names, 414
set_hyperparam_value, 414
set_hyperparams, 414
set_image_in_stack, 415
set_intensity_mcmc_sampling, 415
set_lbound, 415
set_max_sigma, 415
set_max_sigma_ratio, 415
set_mcmc_num_phases, 415
set_mcmc_sigma_scale, 416
set_min_sigma, 416
set_param_names, 416
set_prior, 416
set_rng_seed, 416
set_size, 416
set_ubound, 417
sigma_scale, 423
size, 423
StencilVecT, 399
theta_in_bounds, 417
theta_stack_in_bounds, 417
ubound, 424
update_internal_1Dsum_estimators, 417
x_model, 424
y_model, 424
mappel::Gauss2DsModel, 425
bound_theta, 432
bounded_theta, 432
bounded_theta_stack, 432
bounds_epsilon, 449
check_image_shape, 433
check_param_shape, 433
check_psf_sigma, 433
check_size, 434
compute_max_sigma_ratio, 434
debug_internal_sum_model_x, 434
debug_internal_sum_model_y, 434
default_alpha_sigma, 449
default_beta_pos, 449
default_intensity_kappa, 449
default_max_l, 449
default_mean_l, 449
default_pixel_mean_bg, 450
default_sigma_pos, 450
DefaultPriorType, 450
DefaultSeperableInitEstimator, 450
eta_bg, 450
eta_l, 450
eta_sigma, 451
eta_x, 451
eta_y, 451
Gauss1DSumModelT, 430
Gauss2DsModel, 432
get_hyperparam_index, 434
get_hyperparam_names, 434
get_hyperparam_value, 435
get_hyperparams, 435
get_image_from_stack, 435
get_lbound, 435
get_max_sigma, 435
get_max_sigma_ratio, 436
get_mcmc_num_phases, 436
get_mcmc_sigma_scale, 436

[get_min_sigma](#), 436
[get_num_hyperparams](#), 436
[get_num_params](#), 436
[get_num_pixels](#), 436
[get_param_names](#), 437
[get_prior](#), 437
[get_rng_generator](#), 437
[get_rng_manager](#), 437
[get_size](#), 437
[get_size_image_stack](#), 438
[get_stats](#), 438
[get_ubound](#), 438
[global_default_mcmc_sigma_scale](#), 451
[global_max_mcmc_sigma_scale](#), 451
[global_max_psf_sigma](#), 452
[global_max_size](#), 452
[global_min_psf_sigma](#), 452
[global_min_size](#), 452
[has_hyperparam](#), 438
[ImageCoordT](#), 430
[ImagePixelT](#), 430
[ImageShapeT](#), 430
[ImageSizeShapeT](#), 430
[ImageSizeVecShapeT](#), 430
[ImageSizeVecT](#), 431
[ImageSizeT](#), 430
[ImageStackShapeT](#), 431
[ImageStackT](#), 431
[ImageT](#), 431
[initial_theta_estimate](#), 438, 439
[lbound](#), 452
[make_default_prior](#), 439
[make_default_prior_beta_position](#), 439
[make_default_prior_normal_position](#), 439
[make_image](#), 439
[make_image_stack](#), 440
[make_internal_1Dsum_estimator](#), 440
[make_param](#), 440
[make_param_mat](#), 440
[make_param_mat_stack](#), 441
[make_param_stack](#), 441
[make_prior_beta_position](#), 441
[make_prior_component_intensity](#), 441
[make_prior_component_position_beta](#), 442
[make_prior_component_position_normal](#), 442
[make_prior_component_sigma](#), 442
[make_prior_normal_position](#), 442
[make_stencil](#), 442
[min_sigma](#), 452
[num_dim](#), 453
[num_hyperparams](#), 453
[num_params](#), 453
[num_phases](#), 453
[operator=](#), 443
[ParamVecT](#), 431
[ParamT](#), 431
[pixel_grad](#), 443
[pixel_grad2](#), 443
[pixel_hess](#), 444
[pixel_hess_update](#), 444
[pixel_model_value](#), 444
[prior](#), 453
[prior_types](#), 454
[reflected_theta](#), 444
[reflected_theta_stack](#), 444
[rename_hyperparam](#), 445
[sample_mcmc_candidate](#), 445
[sample_prior](#), 445
[set_background_mcmc_sampling](#), 445
[set_bounds](#), 445
[set_hyperparam_names](#), 446
[set_hyperparam_value](#), 446
[set_hyperparams](#), 446
[set_image_in_stack](#), 446
[set_intensity_mcmc_sampling](#), 446
[set_lbound](#), 446
[set_max_sigma](#), 447
[set_max_sigma_ratio](#), 447
[set_mcmc_num_phases](#), 447
[set_mcmc_sigma_scale](#), 447
[set_min_sigma](#), 447
[set_param_names](#), 447
[set_prior](#), 448
[set_rng_seed](#), 448
[set_size](#), 448
[set_ubound](#), 448
[sigma_scale](#), 454
[size](#), 454
[StencilVecT](#), 431
[theta_in_bounds](#), 448
[theta_stack_in_bounds](#), 448
[ubound](#), 455
[update_internal_1Dsum_estimators](#), 449
[x_model](#), 455
[y_model](#), 455
[mappel::Gauss2DsModel::Stencil](#), 747
[bg](#), 749
[compute_derivatives](#), 749
[DXS2](#), 751
[DXSX](#), 751
[DXS](#), 751
[DYS2](#), 752
[DYSY](#), 752
[DYS](#), 751
[derivatives_computed](#), 750
[DX](#), 751
[dx](#), 750
[DY](#), 751

- dy, [751](#)
- Gx, [752](#)
- Gy, [752](#)
- I, [749](#)
- model, [752](#)
- operator<<, [750](#)
- ParamT, [748](#)
- sigma_ratio, [749](#)
- sigmaX, [749](#)
- sigmaY, [750](#)
- Stencil, [748](#)
- theta, [752](#)
- X, [752](#)
- x, [750](#)
- Y, [753](#)
- y, [750](#)
- mappel::Gauss2DsxyMAP, [456](#)
 - bound_theta, [462](#)
 - bounded_theta, [462](#)
 - bounded_theta_stack, [463](#)
 - bounds_epsilon, [477](#)
 - check_image_shape, [463](#)
 - check_param_shape, [463](#)
 - check_psf_sigma, [463](#), [464](#)
 - check_size, [464](#)
 - default_alpha_sigma, [477](#)
 - default_beta_pos, [477](#)
 - default_intensity_kappa, [477](#)
 - default_max_I, [477](#)
 - default_mean_I, [478](#)
 - default_pixel_mean_bg, [478](#)
 - default_sigma_pos, [478](#)
 - DefaultPriorType, [478](#)
 - DefaultSeperableInitEstimator, [478](#)
 - estimator_names, [478](#)
 - eta_bg, [479](#)
 - eta_I, [479](#)
 - eta_x, [479](#)
 - Gauss2DsxyMAP, [462](#)
 - get_hyperparam_index, [464](#)
 - get_hyperparam_names, [464](#)
 - get_hyperparam_value, [464](#)
 - get_hyperparams, [464](#)
 - get_image_from_stack, [465](#)
 - get_lbound, [465](#)
 - get_mcmc_num_phases, [465](#)
 - get_mcmc_sigma_scale, [465](#)
 - get_num_hyperparams, [465](#)
 - get_num_params, [465](#)
 - get_num_pixels, [465](#)
 - get_param_names, [466](#)
 - get_prior, [466](#)
 - get_psf_sigma, [466](#)
 - get_rng_generator, [466](#)
 - get_rng_manager, [466](#)
 - get_size, [467](#)
 - get_size_image_stack, [467](#)
 - get_stats, [467](#)
 - get_ubound, [467](#)
 - global_default_mcmc_sigma_scale, [479](#)
 - global_max_mcmc_sigma_scale, [479](#)
 - global_max_psf_sigma, [479](#)
 - global_max_size, [480](#)
 - global_min_psf_sigma, [480](#)
 - global_min_size, [480](#)
 - has_hyperparam, [467](#)
 - ImageCoordT, [460](#)
 - ImagePixelT, [460](#)
 - ImageShapeT, [460](#)
 - ImageSizeShapeT, [460](#)
 - ImageSizeVecShapeT, [461](#)
 - ImageSizeVecT, [461](#)
 - ImageSizeT, [461](#)
 - ImageStackShapeT, [461](#)
 - ImageStackT, [461](#)
 - ImageT, [461](#)
 - initial_theta_estimate, [467](#), [468](#)
 - lbound, [480](#)
 - make_default_prior, [468](#)
 - make_default_prior_beta_position, [468](#)
 - make_default_prior_normal_position, [468](#)
 - make_image, [468](#)
 - make_image_stack, [468](#)
 - make_param, [469](#)
 - make_param_mat, [469](#)
 - make_param_mat_stack, [469](#)
 - make_param_stack, [469](#), [470](#)
 - make_prior_beta_position, [470](#)
 - make_prior_component_intensity, [470](#)
 - make_prior_component_position_beta, [470](#)
 - make_prior_component_position_normal, [470](#)
 - make_prior_component_sigma, [471](#)
 - make_prior_normal_position, [471](#)
 - make_stencil, [471](#)
 - ModelDataStackT, [461](#)
 - ModelDataT, [462](#)
 - name, [480](#)
 - num_dim, [480](#)
 - num_hyperparams, [481](#)
 - num_params, [481](#)
 - num_phases, [481](#)
 - ParamVecT, [462](#)
 - ParamT, [462](#)
 - pixel_grad, [472](#)
 - pixel_grad2, [472](#)
 - pixel_hess, [472](#)
 - pixel_hess_update, [472](#)
 - pixel_model_value, [472](#)

prior, 481
 prior_types, 481
 psf_sigma, 482
 reflected_theta, 472
 reflected_theta_stack, 473
 rename_hyperparam, 473
 sample_mcmc_candidate, 473
 sample_prior, 473
 set_background_mcmc_sampling, 473
 set_bounds, 474
 set_hyperparam_names, 474
 set_hyperparam_value, 474
 set_hyperparams, 474
 set_image_in_stack, 474
 set_intensity_mcmc_sampling, 474
 set_lbound, 475
 set_mcmc_num_phases, 475
 set_mcmc_sigma_scale, 475
 set_param_names, 475
 set_prior, 475
 set_psf_sigma, 476
 set_rng_seed, 476
 set_size, 476
 set_ubound, 476
 sigma_scale, 482
 size, 482
 StencilVecT, 462
 theta_in_bounds, 476
 theta_stack_in_bounds, 477
 ubound, 482
 mappel::Gauss2DsxyModel, 483
 bound_theta, 489
 bounded_theta, 489
 bounded_theta_stack, 489
 bounds_epsilon, 501
 check_image_shape, 489
 check_param_shape, 489, 490
 check_psf_sigma, 490
 check_size, 490
 compute_max_sigma_ratio, 490
 default_alpha_sigma, 501
 default_beta_pos, 501
 default_intensity_kappa, 501
 default_max_l, 501
 default_mean_l, 502
 default_pixel_mean_bg, 502
 default_sigma_pos, 502
 DefaultSeperableInitEstimator, 502
 Gauss2DsxyModel, 489
 get_hyperparam_index, 491
 get_hyperparam_names, 491
 get_hyperparam_value, 491
 get_hyperparams, 491
 get_image_from_stack, 491
 get_lbound, 491
 get_max_sigma, 491, 492
 get_max_sigma_ratio, 492
 get_min_sigma, 492
 get_num_hyperparams, 492
 get_num_params, 492
 get_num_pixels, 492
 get_param_names, 492
 get_prior, 493
 get_rng_generator, 493
 get_rng_manager, 493
 get_size, 493
 get_size_image_stack, 493
 get_stats, 494
 get_ubound, 494
 global_max_psf_sigma, 502
 global_max_size, 502
 global_min_psf_sigma, 503
 global_min_size, 503
 has_hyperparam, 494
 ImageCoordT, 487
 ImagePixelT, 487
 ImageShapeT, 487
 ImageSizeShapeT, 487
 ImageSizeVecShapeT, 487
 ImageSizeVecT, 488
 ImageSizeT, 487
 ImageStackShapeT, 488
 ImageStackT, 488
 ImageT, 488
 initial_theta_estimate, 494
 lbound, 503
 make_default_prior, 494
 make_image, 494
 make_image_stack, 494
 make_param, 495
 make_param_mat, 495
 make_param_mat_stack, 495
 make_param_stack, 495, 496
 make_prior_beta_position, 496
 make_prior_component_intensity, 496
 make_prior_component_position_beta, 496
 make_prior_component_position_normal, 496
 make_prior_component_sigma, 496
 make_prior_normal_position, 497
 make_stencil, 497
 mcmc_candidate_eta_sigma, 503
 mcmc_candidate_eta_y, 503
 min_sigma, 503
 num_dim, 504
 num_hyperparams, 504
 num_params, 504
 ParamVecT, 488
 ParamT, 488

- pixel_grad, [497](#)
- pixel_grad2, [497](#)
- pixel_hess, [498](#)
- pixel_hess_update, [498](#)
- pixel_model_value, [498](#)
- prior, [504](#)
- reflected_theta, [498](#)
- reflected_theta_stack, [498](#)
- rename_hyperparam, [498](#)
- sample_mcmc_candidate, [499](#)
- sample_prior, [499](#)
- set_bounds, [499](#)
- set_hyperparam_names, [499](#)
- set_hyperparam_value, [499](#)
- set_hyperparams, [499](#)
- set_image_in_stack, [500](#)
- set_lbound, [500](#)
- set_max_sigma, [500](#)
- set_max_sigma_ratio, [500](#)
- set_min_sigma, [500](#)
- set_param_names, [500](#)
- set_prior, [500](#)
- set_rng_seed, [500](#)
- set_size, [500](#)
- set_ubound, [500](#)
- size, [504](#)
- StencilVecT, [488](#)
- theta_in_bounds, [500](#)
- theta_stack_in_bounds, [501](#)
- ubound, [505](#)
- update_internal_1D_estimators, [501](#)
- x_model, [505](#)
- y_model, [505](#)
- mappel::Gauss2DsxyModel::Stencil, [742](#)
 - bg, [744](#)
 - compute_derivatives, [744](#)
 - DXS2, [745](#)
 - DXSX, [745](#)
 - DXS, [745](#)
 - DYS2, [746](#)
 - DYSX, [746](#)
 - DYSY, [746](#)
 - DYS, [746](#)
 - derivatives_computed, [745](#)
 - DX, [745](#)
 - dx, [745](#)
 - DY, [746](#)
 - dy, [746](#)
 - Gx, [746](#)
 - Gy, [746](#)
 - I, [744](#)
 - model, [746](#)
 - operator<<, [745](#)
 - ParamT, [743](#)
 - sigmaX, [744](#)
 - sigmaY, [744](#)
 - Stencil, [744](#)
 - theta, [747](#)
 - X, [747](#)
 - x, [744](#)
 - Y, [747](#)
 - y, [745](#)
- mappel::HeuristicEstimator
 - clear_stats, [507](#)
 - compute_estimate, [507](#)
 - compute_estimate_debug, [507](#)
 - compute_profile_estimate, [508](#)
 - estimate_max, [508](#), [509](#)
 - estimate_max_debug, [509](#)
 - estimate_max_stack, [509](#)
 - estimate_profile_stack, [509](#)
 - get_debug_stats, [510](#)
 - get_model, [510](#)
 - get_stats, [510](#)
 - HeuristicEstimator, [507](#)
 - max_threads, [511](#)
 - model, [511](#)
 - mtx, [511](#)
 - name, [510](#)
 - num_estimations, [511](#)
 - num_threads, [511](#)
 - record_walltime, [510](#)
 - set_model, [511](#)
 - total_walltime, [512](#)
- mappel::HeuristicEstimator< Model >, [506](#)
- mappel::ImageFormat1DBase, [512](#)
 - check_image_shape, [515](#)
 - check_size, [515](#)
 - get_image_from_stack, [515](#)
 - get_num_pixels, [516](#)
 - get_size, [516](#)
 - get_size_image_stack, [516](#)
 - get_stats, [516](#)
 - global_max_size, [517](#)
 - global_min_size, [517](#)
 - ImageCoordT, [514](#)
 - ImageFormat1DBase, [515](#)
 - ImagePixelT, [514](#)
 - ImageShapeT, [514](#)
 - ImageSizeShapeT, [514](#)
 - ImageSizeVecShapeT, [514](#)
 - ImageSizeVecT, [514](#)
 - ImageSizeT, [514](#)
 - ImageStackShapeT, [514](#)
 - ImageStackT, [514](#)
 - ImageT, [515](#)
 - make_image, [516](#)
 - make_image_stack, [516](#)

- num_dim, 517
- set_image_in_stack, 517
- set_size, 517
- size, 517
- mappel::ImageFormat2DBase, 518
 - check_image_shape, 522
 - check_size, 522
 - get_image_from_stack, 522
 - get_num_pixels, 522
 - get_size, 522
 - get_size_image_stack, 523
 - get_stats, 523
 - global_max_size, 524
 - global_min_size, 524
 - ImageCoordT, 520
 - ImageFormat2DBase, 521
 - ImagePixelT, 520
 - ImageShapeT, 520
 - ImageSizeShapeT, 520
 - ImageSizeVecShapeT, 520
 - ImageSizeVecT, 520
 - ImageSizeT, 520
 - ImageStackShapeT, 521
 - ImageStackT, 521
 - ImageT, 521
 - make_image, 523
 - make_image_stack, 523
 - num_dim, 524
 - operator=, 523
 - set_image_in_stack, 523
 - set_size, 524
 - size, 524
- mappel::IterativeMaximizer
 - alpha, 534
 - backtrack, 528
 - clear_stats, 528
 - compute_estimate, 529
 - compute_estimate_debug, 529
 - compute_profile_estimate, 529
 - convergence_test, 530
 - delta, 534
 - epsilon, 534
 - Error, 528
 - estimate_max, 530, 531
 - estimate_max_debug, 531
 - estimate_max_stack, 531
 - estimate_profile_stack, 531
 - exit_counts, 534
 - ExitCode, 528
 - FunctionChange, 528
 - get_debug_stats, 532
 - get_model, 532
 - get_stats, 532
 - GradRatio, 528
 - IterativeMaximizer, 528
 - lambda_min, 534
 - last_backtrack_idx, 535
 - local_maximize, 532
 - max_backtracks, 535
 - max_iterations, 535
 - max_threads, 535
 - MaxBacktracks, 528
 - MaxIter, 528
 - maximize, 533
 - mean_backtracks, 533
 - mean_der_evals, 533
 - mean_fun_evals, 533
 - mean_iterations, 533
 - model, 535
 - mtx, 535
 - name, 533
 - num_estimations, 536
 - num_threads, 536
 - NumExitCodes, 536
 - record_run_statistics, 533
 - record_walltime, 533
 - set_model, 534
 - StepSize, 528
 - total_backtracks, 536
 - total_der_evals, 536
 - total_fun_evals, 536
 - total_iterations, 536
 - total_walltime, 537
 - TrustRegionRadius, 528
 - Unassigned, 528
- mappel::IterativeMaximizer< Model >, 525
- mappel::IterativeMaximizer< Model >::MaximizerData, 553
- mappel::IterativeMaximizer::MaximizerData
 - backtrack_idx, 558
 - current_stencil, 558
 - exit_code, 558
 - fixed_parameter_scalar, 558
 - get_backtrack_idx, 555
 - get_theta_sequence, 555
 - get_theta_sequence_rllh, 555
 - getIteration, 555
 - grad, 558
 - has_fixed_parameters, 558
 - im, 558
 - lbound, 558
 - max_seq_len, 559
 - MaximizerData, 554
 - nBacktracks, 559
 - nIterations, 559
 - record_backtrack, 555
 - record_exit, 555
 - record_iteration, 556

- restore_stencil, 556
- rllh, 559
- s0, 559
- s1, 559
- save_seq, 559
- save_stencil, 556
- saved_stencil, 556
- saved_theta, 557
- seq_len, 560
- seq_rllh, 560
- set_fixed_parameters, 557
- set_stencil, 557
- stencil, 557
- step, 560
- theta, 557
- theta_seq, 560
- ubound, 560
- mappel::LogicalError, 537
 - LogicalError, 538
- mappel::MAPEstimator, 538
 - bound_theta, 541
 - bounded_theta, 541
 - bounded_theta_stack, 541
 - bounds_epsilon, 550
 - check_param_shape, 541, 542
 - check_psf_sigma, 542
 - default_alpha_sigma, 550
 - default_beta_pos, 550
 - default_intensity_kappa, 550
 - default_max_l, 550
 - default_mean_l, 550
 - default_pixel_mean_bg, 550
 - default_sigma_pos, 551
 - DefaultSeperableInitEstimator, 551
 - get_hyperparam_index, 542
 - get_hyperparam_names, 542
 - get_hyperparam_value, 542
 - get_hyperparams, 543
 - get_lbound, 543
 - get_num_hyperparams, 543
 - get_num_params, 543
 - get_param_names, 543
 - get_prior, 543
 - get_rng_generator, 544
 - get_rng_manager, 544
 - get_stats, 544
 - get_ubound, 544
 - global_max_psf_sigma, 551
 - global_min_psf_sigma, 551
 - has_hyperparam, 544
 - lbound, 551
 - MAPEstimator, 541
 - make_param, 544, 545
 - make_param_mat, 545
 - make_param_mat_stack, 545
 - make_param_stack, 545
 - make_prior_component_intensity, 546
 - make_prior_component_position_beta, 546
 - make_prior_component_position_normal, 546
 - make_prior_component_sigma, 546
 - num_hyperparams, 552
 - num_params, 552
 - operator=, 546, 547
 - ParamVecT, 540
 - ParamT, 540
 - prior, 552
 - reflected_theta, 547
 - reflected_theta_stack, 547
 - rename_hyperparam, 547
 - sample_prior, 547
 - set_bounds, 547
 - set_hyperparam_names, 548
 - set_hyperparam_value, 548
 - set_hyperparams, 548
 - set_lbound, 548
 - set_param_names, 548
 - set_prior, 548, 549
 - set_rng_seed, 549
 - set_ubound, 549
 - theta_in_bounds, 549
 - theta_stack_in_bounds, 549
 - ubound, 552
- mappel::MCMCAdaptor1Ds, 578
 - bound_theta, 582
 - bounded_theta, 582
 - bounded_theta_stack, 582
 - bounds_epsilon, 591
 - check_param_shape, 582
 - check_psf_sigma, 583
 - default_alpha_sigma, 591
 - default_beta_pos, 592
 - default_intensity_kappa, 592
 - default_max_l, 592
 - default_mean_l, 592
 - default_pixel_mean_bg, 592
 - default_sigma_pos, 592
 - DefaultSeperableInitEstimator, 593
 - eta_bg, 593
 - eta_l, 593
 - eta_sigma, 593
 - eta_x, 593
 - get_hyperparam_index, 583
 - get_hyperparam_names, 583
 - get_hyperparam_value, 583
 - get_hyperparams, 583
 - get_lbound, 584
 - get_mcmc_num_phases, 584
 - get_mcmc_sigma_scale, 584

get_num_hyperparams, 584
get_num_params, 584
get_param_names, 584
get_prior, 584, 585
get_rng_generator, 585
get_rng_manager, 585
get_stats, 585
get_ubound, 585
global_default_mcmc_sigma_scale, 594
global_max_mcmc_sigma_scale, 594
global_max_psf_sigma, 594
global_min_psf_sigma, 594
has_hyperparam, 585
lbound, 594
MCMCAdaptor1Ds, 581, 582
make_param, 585, 586
make_param_mat, 586
make_param_mat_stack, 586
make_param_stack, 586
make_prior_component_intensity, 587
make_prior_component_position_beta, 587
make_prior_component_position_normal, 587
make_prior_component_sigma, 587
num_hyperparams, 594
num_params, 595
num_phases, 595
operator=, 587, 588
ParamVecT, 581
ParamT, 581
prior, 595
reflected_theta, 588
reflected_theta_stack, 588
rename_hyperparam, 588
sample_mcmc_candidate, 588
sample_prior, 588
set_background_mcmc_sampling, 589
set_bounds, 589
set_hyperparam_names, 589
set_hyperparam_value, 589
set_hyperparams, 589
set_intensity_mcmc_sampling, 589
set_lbound, 590
set_mcmc_num_phases, 590
set_mcmc_sigma_scale, 590
set_param_names, 590
set_prior, 590
set_rng_seed, 591
set_ubound, 591
sigma_scale, 595
theta_in_bounds, 591
theta_stack_in_bounds, 591
ubound, 596
mappel::MCMCAdaptor1D, 561
bound_theta, 564
bounded_theta, 564
bounded_theta_stack, 564
bounds_epsilon, 574
check_param_shape, 565
check_psf_sigma, 565
default_alpha_sigma, 574
default_beta_pos, 574
default_intensity_kappa, 574
default_max_l, 574
default_mean_l, 575
default_pixel_mean_bg, 575
default_sigma_pos, 575
DefaultSeperableInitEstimator, 575
eta_bg, 575
eta_l, 575
eta_x, 576
get_hyperparam_index, 565
get_hyperparam_names, 565
get_hyperparam_value, 566
get_hyperparams, 566
get_lbound, 566
get_mcmc_num_phases, 566
get_mcmc_sigma_scale, 566
get_num_hyperparams, 566
get_num_params, 566
get_param_names, 567
get_prior, 567
get_rng_generator, 567
get_rng_manager, 567
get_stats, 567
get_ubound, 567
global_default_mcmc_sigma_scale, 576
global_max_mcmc_sigma_scale, 576
global_max_psf_sigma, 576
global_min_psf_sigma, 576
has_hyperparam, 568
lbound, 576
MCMCAdaptor1D, 564
make_param, 568
make_param_mat, 568
make_param_mat_stack, 568, 569
make_param_stack, 569
make_prior_component_intensity, 569
make_prior_component_position_beta, 569
make_prior_component_position_normal, 569
make_prior_component_sigma, 570
num_hyperparams, 577
num_params, 577
num_phases, 577
operator=, 570
ParamVecT, 563
ParamT, 563
prior, 577
reflected_theta, 570

- reflected_theta_stack, 570
- rename_hyperparam, 570
- sample_mcmc_candidate, 571
- sample_prior, 571
- set_background_mcmc_sampling, 571
- set_bounds, 571
- set_hyperparam_names, 571
- set_hyperparam_value, 572
- set_hyperparams, 572
- set_intensity_mcmc_sampling, 572
- set_lbound, 572
- set_mcmc_num_phases, 572
- set_mcmc_sigma_scale, 572
- set_param_names, 573
- set_prior, 573
- set_rng_seed, 573
- set_ubound, 573
- sigma_scale, 577
- theta_in_bounds, 573
- theta_stack_in_bounds, 574
- ubound, 578
- mappel::MCMCAdaptor2Ds, 614
 - bound_theta, 618
 - bounded_theta, 618
 - bounded_theta_stack, 618
 - bounds_epsilon, 628
 - check_param_shape, 618
 - check_psf_sigma, 619
 - default_alpha_sigma, 628
 - default_beta_pos, 628
 - default_intensity_kappa, 628
 - default_max_l, 628
 - default_mean_l, 628
 - default_pixel_mean_bg, 628
 - default_sigma_pos, 629
 - DefaultSeperableInitEstimator, 629
 - eta_bg, 629
 - eta_l, 629
 - eta_sigma, 629
 - eta_x, 630
 - eta_y, 630
 - get_hyperparam_index, 619
 - get_hyperparam_names, 619
 - get_hyperparam_value, 619
 - get_hyperparams, 619
 - get_lbound, 620
 - get_mcmc_num_phases, 620
 - get_mcmc_sigma_scale, 620
 - get_num_hyperparams, 620
 - get_num_params, 620
 - get_param_names, 620
 - get_prior, 620, 621
 - get_rng_generator, 621
 - get_rng_manager, 621
 - get_stats, 621
 - get_ubound, 621
 - global_default_mcmc_sigma_scale, 630
 - global_max_mcmc_sigma_scale, 630
 - global_max_psf_sigma, 630
 - global_min_psf_sigma, 630
 - has_hyperparam, 621
 - lbound, 631
 - MCMCAdaptor2Ds, 617, 618
 - make_param, 621, 622
 - make_param_mat, 622
 - make_param_mat_stack, 622
 - make_param_stack, 622
 - make_prior_component_intensity, 623
 - make_prior_component_position_beta, 623
 - make_prior_component_position_normal, 623
 - make_prior_component_sigma, 623
 - num_hyperparams, 631
 - num_params, 631
 - num_phases, 631
 - operator=, 623, 624
 - ParamVecT, 617
 - ParamT, 617
 - prior, 631
 - reflected_theta, 624
 - reflected_theta_stack, 624
 - rename_hyperparam, 624
 - sample_mcmc_candidate, 624
 - sample_prior, 624, 625
 - set_background_mcmc_sampling, 625
 - set_bounds, 625
 - set_hyperparam_names, 625
 - set_hyperparam_value, 625
 - set_hyperparams, 625
 - set_intensity_mcmc_sampling, 626
 - set_lbound, 626
 - set_mcmc_num_phases, 626
 - set_mcmc_sigma_scale, 626
 - set_param_names, 626
 - set_prior, 626, 627
 - set_rng_seed, 627
 - set_ubound, 627
 - sigma_scale, 632
 - theta_in_bounds, 627
 - theta_stack_in_bounds, 627
 - ubound, 632
- mappel::MCMCAdaptor2D, 596
 - bound_theta, 600
 - bounded_theta, 600
 - bounded_theta_stack, 600
 - bounds_epsilon, 609
 - check_param_shape, 600
 - check_psf_sigma, 601
 - default_alpha_sigma, 609

default_beta_pos, 610
 default_intensity_kappa, 610
 default_max_l, 610
 default_mean_l, 610
 default_pixel_mean_bg, 610
 default_sigma_pos, 610
 DefaultSeperableInitEstimator, 611
 eta_bg, 611
 eta_l, 611
 eta_x, 611
 eta_y, 611
 get_hyperparam_index, 601
 get_hyperparam_names, 601
 get_hyperparam_value, 601
 get_hyperparams, 601
 get_lbound, 602
 get_mcmc_num_phases, 602
 get_mcmc_sigma_scale, 602
 get_num_hyperparams, 602
 get_num_params, 602
 get_param_names, 602
 get_prior, 602, 603
 get_rng_generator, 603
 get_rng_manager, 603
 get_stats, 603
 get_ubound, 603
 global_default_mcmc_sigma_scale, 612
 global_max_mcmc_sigma_scale, 612
 global_max_psf_sigma, 612
 global_min_psf_sigma, 612
 has_hyperparam, 603
 lbound, 612
 MCMCAdaptor2D, 599, 600
 make_param, 603, 604
 make_param_mat, 604
 make_param_mat_stack, 604
 make_param_stack, 604
 make_prior_component_intensity, 605
 make_prior_component_position_beta, 605
 make_prior_component_position_normal, 605
 make_prior_component_sigma, 605
 num_hyperparams, 612
 num_params, 613
 num_phases, 613
 operator=, 605, 606
 ParamVecT, 599
 ParamT, 599
 prior, 613
 reflected_theta, 606
 reflected_theta_stack, 606
 rename_hyperparam, 606
 sample_mcmc_candidate, 606
 sample_prior, 606
 set_background_mcmc_sampling, 607
 set_bounds, 607
 set_hyperparam_names, 607
 set_hyperparam_value, 607
 set_hyperparams, 607
 set_intensity_mcmc_sampling, 607
 set_lbound, 608
 set_mcmc_num_phases, 608
 set_mcmc_sigma_scale, 608
 set_param_names, 608
 set_prior, 608
 set_rng_seed, 609
 set_ubound, 609
 sigma_scale, 613
 theta_in_bounds, 609
 theta_stack_in_bounds, 609
 ubound, 614
 mappel::MCMCAdaptorBase, 633
 get_mcmc_num_phases, 634
 get_mcmc_sigma_scale, 634
 get_stats, 634
 global_default_mcmc_sigma_scale, 635
 global_max_mcmc_sigma_scale, 635
 MCMCAdaptorBase, 633
 num_phases, 635
 set_mcmc_num_phases, 634
 set_mcmc_sigma_scale, 634
 sigma_scale, 635
 mappel::MLEstimator, 635
 bound_theta, 638
 bounded_theta, 638
 bounded_theta_stack, 639
 bounds_epsilon, 647
 check_param_shape, 639
 check_psf_sigma, 639
 default_alpha_sigma, 647
 default_beta_pos, 647
 default_intensity_kappa, 647
 default_max_l, 647
 default_mean_l, 647
 default_pixel_mean_bg, 647
 default_sigma_pos, 648
 DefaultSeperableInitEstimator, 648
 get_hyperparam_index, 639
 get_hyperparam_names, 640
 get_hyperparam_value, 640
 get_hyperparams, 640
 get_lbound, 640
 get_num_hyperparams, 640
 get_num_params, 640
 get_param_names, 640
 get_prior, 641
 get_rng_generator, 641
 get_rng_manager, 641
 get_stats, 641

- get_ubound, [641](#)
- global_max_psf_sigma, [648](#)
- global_min_psf_sigma, [648](#)
- has_hyperparam, [641](#)
- lbound, [648](#)
- MLEstimator, [638](#)
- make_param, [642](#)
- make_param_mat, [642](#)
- make_param_mat_stack, [642](#)
- make_param_stack, [642](#), [643](#)
- make_prior_component_intensity, [643](#)
- make_prior_component_position_beta, [643](#)
- make_prior_component_position_normal, [643](#)
- make_prior_component_sigma, [643](#)
- num_hyperparams, [649](#)
- num_params, [649](#)
- operator=, [644](#)
- ParamVecT, [638](#)
- ParamT, [638](#)
- prior, [649](#)
- reflected_theta, [644](#)
- reflected_theta_stack, [644](#)
- rename_hyperparam, [644](#)
- sample_prior, [644](#)
- set_bounds, [645](#)
- set_hyperparam_names, [645](#)
- set_hyperparam_value, [645](#)
- set_hyperparams, [645](#)
- set_lbound, [645](#)
- set_param_names, [645](#)
- set_prior, [646](#)
- set_rng_seed, [646](#)
- set_ubound, [646](#)
- theta_in_bounds, [646](#)
- theta_stack_in_bounds, [646](#)
- ubound, [649](#)
- mappel::ModelError, [650](#)
 - ModelError, [650](#)
- mappel::NewtonDiagonalMaximizer
 - alpha, [660](#)
 - backtrack, [654](#)
 - clear_stats, [654](#)
 - compute_estimate, [654](#)
 - compute_estimate_debug, [655](#)
 - compute_profile_estimate, [655](#)
 - convergence_test, [655](#)
 - delta, [660](#)
 - epsilon, [660](#)
 - Error, [653](#)
 - estimate_max, [656](#)
 - estimate_max_debug, [657](#)
 - estimate_max_stack, [657](#)
 - estimate_profile_stack, [657](#)
 - exit_counts, [660](#)
 - ExitCode, [653](#)
 - FunctionChange, [653](#)
 - get_debug_stats, [657](#)
 - get_model, [658](#)
 - get_stats, [658](#)
 - GradRatio, [653](#)
 - lambda_min, [660](#)
 - last_backtrack_idx, [661](#)
 - local_maximize, [658](#)
 - max_backtracks, [661](#)
 - max_iterations, [661](#)
 - max_threads, [661](#)
 - MaxBacktracks, [653](#)
 - MaxIter, [653](#)
 - maximize, [658](#), [659](#)
 - MaximizerData, [653](#)
 - mean_backtracks, [659](#)
 - mean_der_evals, [659](#)
 - mean_fun_evals, [659](#)
 - mean_iterations, [659](#)
 - model, [661](#)
 - mtx, [661](#)
 - name, [659](#)
 - NewtonDiagonalMaximizer, [653](#)
 - num_estimations, [662](#)
 - num_threads, [662](#)
 - NumExitCodes, [662](#)
 - record_run_statistics, [659](#)
 - record_walltime, [659](#)
 - set_model, [660](#)
 - StepSize, [653](#)
 - total_backtracks, [662](#)
 - total_der_evals, [662](#)
 - total_fun_evals, [662](#)
 - total_iterations, [663](#)
 - total_walltime, [663](#)
 - TrustRegionRadius, [653](#)
 - Unassigned, [653](#)
- mappel::NewtonDiagonalMaximizer< Model >, [651](#)
- mappel::NewtonMaximizer
 - alpha, [672](#)
 - backtrack, [666](#)
 - clear_stats, [666](#)
 - compute_estimate, [667](#)
 - compute_estimate_debug, [667](#)
 - compute_profile_estimate, [667](#)
 - convergence_test, [668](#)
 - delta, [672](#)
 - epsilon, [672](#)
 - Error, [666](#)
 - estimate_max, [668](#), [669](#)
 - estimate_max_debug, [669](#)
 - estimate_max_stack, [669](#)
 - estimate_profile_stack, [669](#)

exit_counts, 673
 ExitCode, 666
 FunctionChange, 666
 get_debug_stats, 670
 get_model, 670
 get_stats, 670
 GradRatio, 666
 lambda_min, 673
 last_backtrack_idx, 673
 local_maximize, 670
 max_backtracks, 673
 max_iterations, 673
 max_threads, 673
 MaxBacktracks, 666
 MaxIter, 666
 maximize, 671
 MaximizerData, 666
 mean_backtracks, 671
 mean_der_evals, 671
 mean_fun_evals, 671
 mean_iterations, 671
 model, 674
 mtx, 674
 name, 671
 NewtonMaximizer, 666
 num_estimations, 674
 num_threads, 674
 NumExitCodes, 674
 record_run_statistics, 671
 record_walltime, 672
 set_model, 672
 StepSize, 666
 total_backtracks, 674
 total_der_evals, 675
 total_fun_evals, 675
 total_iterations, 675
 total_walltime, 675
 TrustRegionRadius, 666
 Unassigned, 666
 mappel::NewtonMaximizer< Model >, 663
 mappel::NotImplementedError, 676
 NotImplementedError, 676
 mappel::NumericalError, 676
 NumericalError, 677
 mappel::ParameterValueError, 679
 ParameterValueError, 680
 mappel::PointEmitterModel, 680
 bound_theta, 684
 bounded_theta, 684
 bounded_theta_stack, 684
 bounds_epsilon, 692
 check_param_shape, 684
 check_psf_sigma, 685
 default_alpha_sigma, 692
 default_beta_pos, 692
 default_intensity_kappa, 692
 default_max_l, 693
 default_mean_l, 693
 default_pixel_mean_bg, 693
 default_sigma_pos, 693
 DefaultSeperableInitEstimator, 693
 get_hyperparam_index, 685
 get_hyperparam_names, 685
 get_hyperparam_value, 685
 get_hyperparams, 685
 get_lbound, 685
 get_num_hyperparams, 686
 get_num_params, 686
 get_param_names, 686
 get_prior, 686
 get_rng_generator, 686
 get_rng_manager, 686
 get_stats, 687
 get_ubound, 687
 global_max_psf_sigma, 693
 global_min_psf_sigma, 694
 has_hyperparam, 687
 lbound, 694
 make_param, 687
 make_param_mat, 687, 688
 make_param_mat_stack, 688
 make_param_stack, 688
 make_prior_component_intensity, 688
 make_prior_component_position_beta, 688
 make_prior_component_position_normal, 689
 make_prior_component_sigma, 689
 num_hyperparams, 694
 num_params, 694
 operator=, 689
 ParamVecT, 683
 ParamT, 683
 PointEmitterModel, 683, 684
 prior, 694
 reflected_theta, 689
 reflected_theta_stack, 690
 rename_hyperparam, 690
 sample_prior, 690
 set_bounds, 690
 set_hyperparam_names, 690
 set_hyperparam_value, 690
 set_hyperparams, 691
 set_lbound, 691
 set_param_names, 691
 set_prior, 691
 set_rng_seed, 691
 set_ubound, 691
 theta_in_bounds, 692
 theta_stack_in_bounds, 692

- ubound, 694
- mappel::PoissonNoise1DObjective, 697
 - check_image_shape, 701
 - check_size, 701
 - estimator_names, 703
 - get_image_from_stack, 701
 - get_num_pixels, 702
 - get_size, 702
 - get_size_image_stack, 702
 - get_stats, 702
 - global_max_size, 703
 - global_min_size, 704
 - ImageCoordT, 699
 - ImagePixelT, 699
 - ImageShapeT, 699
 - ImageSizeShapeT, 699
 - ImageSizeVecShapeT, 700
 - ImageSizeVecT, 700
 - ImageSizeT, 699
 - ImageStackShapeT, 700
 - ImageStackT, 700
 - ImageT, 700
 - make_image, 702
 - make_image_stack, 702
 - ModelDataStackT, 700
 - ModelDataT, 700
 - num_dim, 704
 - operator=, 703
 - PoissonNoise1DObjective, 701
 - set_image_in_stack, 703
 - set_size, 703
 - size, 704
- mappel::PoissonNoise2DObjective, 705
 - check_image_shape, 708
 - check_size, 709
 - estimator_names, 711
 - get_image_from_stack, 709
 - get_num_pixels, 709
 - get_size, 709
 - get_size_image_stack, 709
 - get_stats, 709
 - global_max_size, 711
 - global_min_size, 711
 - ImageCoordT, 706
 - ImagePixelT, 706
 - ImageShapeT, 706
 - ImageSizeShapeT, 707
 - ImageSizeVecShapeT, 707
 - ImageSizeVecT, 707
 - ImageSizeT, 707
 - ImageStackShapeT, 707
 - ImageStackT, 707
 - ImageT, 707
 - make_image, 710
 - make_image_stack, 710
 - ModelDataStackT, 708
 - ModelDataT, 708
 - num_dim, 711
 - operator=, 710
 - PoissonNoise2DObjective, 708
 - set_image_in_stack, 710
 - set_size, 710
 - size, 711
- mappel::PriorMAP1DObjective, 712
 - log_prior_l_const, 714
 - log_prior_bg_const, 714
 - log_prior_pos_const, 714
 - ParamMatT, 713
 - ParamT, 713
 - prior_grad2_update, 713
 - prior_grad_update, 713
 - prior_hess_update, 713
 - prior_log_likelihood, 713
 - prior_relative_log_likelihood, 714
 - set_hyperparameters, 714
- mappel::QuasiNewtonMaximizer
 - alpha, 724
 - backtrack, 718
 - clear_stats, 718
 - compute_estimate, 718
 - compute_estimate_debug, 719
 - compute_profile_estimate, 719
 - convergence_test, 719
 - delta, 724
 - epsilon, 724
 - Error, 717
 - estimate_max, 720
 - estimate_max_debug, 721
 - estimate_max_stack, 721
 - estimate_profile_stack, 721
 - exit_counts, 724
 - ExitCode, 717
 - FunctionChange, 717
 - get_debug_stats, 721
 - get_model, 722
 - get_stats, 722
 - GradRatio, 717
 - lambda_min, 724
 - last_backtrack_idx, 725
 - local_maximize, 722
 - max_backtracks, 725
 - max_iterations, 725
 - max_threads, 725
 - MaxBacktracks, 717
 - MaxIter, 717
 - maximize, 722, 723
 - MaximizerData, 717
 - mean_backtracks, 723

- mean_der_evals, [723](#)
- mean_fun_evals, [723](#)
- mean_iterations, [723](#)
- model, [725](#)
- mtx, [725](#)
- name, [723](#)
- num_estimations, [726](#)
- num_threads, [726](#)
- NumExitCodes, [726](#)
- QuasiNewtonMaximizer, [718](#)
- record_run_statistics, [723](#)
- record_walltime, [723](#)
- set_model, [724](#)
- StepSize, [717](#)
- total_backtracks, [726](#)
- total_der_evals, [726](#)
- total_fun_evals, [726](#)
- total_iterations, [727](#)
- total_walltime, [727](#)
- TrustRegionRadius, [717](#)
- Unassigned, [717](#)
- mappel::QuasiNewtonMaximizer< Model >, [715](#)
- mappel::SimulatedAnnealingMaximizer
 - anneal, [729](#)
 - clear_stats, [729](#)
 - compute_estimate, [729](#), [730](#)
 - compute_estimate_debug, [730](#)
 - compute_profile_estimate, [730](#)
 - cooling_rate, [733](#)
 - estimate_max, [730](#), [731](#)
 - estimate_max_debug, [731](#)
 - estimate_max_stack, [731](#)
 - estimate_profile_stack, [732](#)
 - get_debug_stats, [732](#)
 - get_model, [732](#)
 - get_stats, [732](#)
 - max_iterations, [733](#)
 - max_threads, [733](#)
 - model, [733](#)
 - mtx, [733](#)
 - name, [732](#)
 - num_estimations, [734](#)
 - num_threads, [734](#)
 - record_walltime, [733](#)
 - set_model, [733](#)
 - SimulatedAnnealingMaximizer, [729](#)
 - T_init, [734](#)
 - total_walltime, [734](#)
- mappel::SimulatedAnnealingMaximizer< Model >, [727](#)
- mappel::ThreadedEstimator
 - clear_stats, [760](#)
 - compute_estimate, [760](#)
 - compute_estimate_debug, [760](#)
 - compute_profile_estimate, [760](#)
- estimate_max, [761](#)
- estimate_max_debug, [761](#)
- estimate_max_stack, [762](#)
- estimate_profile_stack, [762](#)
- get_debug_stats, [762](#)
- get_model, [762](#)
- get_stats, [763](#)
- max_threads, [763](#)
- model, [763](#)
- mtx, [764](#)
- name, [763](#)
- num_estimations, [764](#)
- num_threads, [764](#)
- record_walltime, [763](#)
- set_model, [763](#)
- ThreadedEstimator, [759](#)
- total_walltime, [764](#)
- mappel::ThreadedEstimator< Model >, [758](#)
- mappel::TrustRegionMaximizer
 - alpha, [776](#)
 - backtrack, [768](#)
 - bound_step, [768](#)
 - boundary_stepback_min_kappa, [776](#)
 - clear_stats, [769](#)
 - compute_D_scale, [769](#)
 - compute_bound_scaling_vec, [769](#)
 - compute_cauchy_point, [769](#)
 - compute_estimate, [769](#), [770](#)
 - compute_estimate_debug, [770](#)
 - compute_initial_trust_radius, [770](#)
 - compute_profile_estimate, [770](#)
 - convergence_test, [771](#)
 - delta, [776](#)
 - delta_decrease, [776](#)
 - delta_decrease_min, [776](#)
 - delta_increase, [776](#)
 - delta_init_max, [776](#)
 - delta_init_min, [776](#)
 - epsilon, [776](#)
 - Error, [768](#)
 - estimate_max, [771](#), [772](#)
 - estimate_max_debug, [772](#)
 - estimate_max_stack, [772](#)
 - estimate_profile_stack, [772](#)
 - exit_counts, [777](#)
 - ExitCode, [768](#)
 - FunctionChange, [768](#)
 - get_debug_stats, [773](#)
 - get_model, [773](#)
 - get_stats, [773](#)
 - GradRatio, [768](#)
 - lambda_min, [777](#)
 - last_backtrack_idx, [777](#)
 - local_maximize, [773](#)

- max_backtracks, [777](#)
- max_iterations, [777](#)
- max_scaling, [777](#)
- max_threads, [778](#)
- MaxBacktracks, [768](#)
- MaxIter, [768](#)
- maximize, [774](#)
- MaximizerData, [767](#)
- mean_backtracks, [774](#)
- mean_der_evals, [774](#)
- mean_fun_evals, [774](#)
- mean_iterations, [774](#)
- min_scaling, [778](#)
- model, [778](#)
- mtx, [778](#)
- name, [774](#)
- num_estimations, [778](#)
- num_threads, [778](#)
- NumExitCodes, [779](#)
- quadratic_model_value, [774](#)
- record_run_statistics, [774](#)
- record_walltime, [775](#)
- rho_cauchy_min, [779](#)
- rho_obj_min, [779](#)
- rho_obj_opt, [779](#)
- set_model, [775](#)
- solve_TR_subproblem, [775](#)
- solve_restricted_step_length_newton, [775](#)
- StepSize, [768](#)
- total_backtracks, [779](#)
- total_der_evals, [779](#)
- total_fun_evals, [779](#)
- total_iterations, [779](#)
- total_walltime, [780](#)
- TrustRegionMaximizer, [768](#)
- TrustRegionRadius, [768](#)
- Unassigned, [768](#)
- mappel::TrustRegionMaximizer< Model >, [765](#)
- 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](#), [56](#)
 - estimate_mcmc_sample, [56](#)
 - expected_information, [56](#), [57](#)
 - likelihood_objective, [57](#)
 - make_estimator, [57](#), [58](#)
 - model_image, [58](#)
 - observed_information, [58](#), [59](#)
 - 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](#)
 - rllh, [64](#), [65](#)
- mappel::methods::likelihood::debug, [65](#)
 - grad_components, [66](#)
 - hessian_components, [66](#)
 - llh_components, [66](#)
 - rllh_components, [66](#), [67](#)
- mappel::methods::objective, [67](#)
 - grad, [68](#), [69](#)
 - grad2, [69](#)
 - hessian, [69](#), [70](#)
 - llh, [70](#), [71](#)
 - negative_definite_hessian, [71](#)
 - rllh, [71](#), [72](#)
- mappel::methods::objective::debug, [72](#)
 - grad_components, [73](#)
 - hessian_components, [73](#)
 - llh_components, [74](#)
 - rllh_components, [74](#)
- mappel::methods::objective::openmp, [75](#)
 - grad_stack, [75](#)
 - hessian_stack, [76](#)
 - llh_stack, [76](#), [77](#)
 - negative_definite_hessian_stack, [77](#)
 - rllh_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, 535
 mappel::NewtonDiagonalMaximizer, 661
 mappel::NewtonMaximizer, 673
 mappel::QuasiNewtonMaximizer, 725
 mappel::TrustRegionMaximizer, 777
 max_iterations
 mappel::CGaussMLE, 99
 mappel::IterativeMaximizer, 535
 mappel::NewtonDiagonalMaximizer, 661
 mappel::NewtonMaximizer, 673
 mappel::QuasiNewtonMaximizer, 725
 mappel::SimulatedAnnealingMaximizer, 733
 mappel::TrustRegionMaximizer, 777
 max_scaling
 mappel::TrustRegionMaximizer, 777
 max_seq_len
 mappel::IterativeMaximizer::MaximizerData, 559
 max_threads
 mappel::CGaussHeuristicEstimator, 92
 mappel::CGaussMLE, 99
 mappel::HeuristicEstimator, 511
 mappel::IterativeMaximizer, 535
 mappel::NewtonDiagonalMaximizer, 661
 mappel::NewtonMaximizer, 673
 mappel::QuasiNewtonMaximizer, 725
 mappel::SimulatedAnnealingMaximizer, 733
 mappel::ThreadedEstimator, 763
 mappel::TrustRegionMaximizer, 778
 MaxBacktracks
 mappel::IterativeMaximizer, 528
 mappel::NewtonDiagonalMaximizer, 653
 mappel::NewtonMaximizer, 666
 mappel::QuasiNewtonMaximizer, 717
 mappel::TrustRegionMaximizer, 768
 MaxIter
 mappel::IterativeMaximizer, 528
 mappel::NewtonDiagonalMaximizer, 653
 mappel::NewtonMaximizer, 666
 mappel::QuasiNewtonMaximizer, 717
 mappel::TrustRegionMaximizer, 768
 maxidx
 mappel, 39
 maximize
 mappel::IterativeMaximizer, 533
 mappel::NewtonDiagonalMaximizer, 658, 659
 mappel::NewtonMaximizer, 671
 mappel::QuasiNewtonMaximizer, 722, 723
 mappel::TrustRegionMaximizer, 774
 MaximizerData
 mappel::IterativeMaximizer::MaximizerData, 554
 mappel::NewtonDiagonalMaximizer, 653
 mappel::NewtonMaximizer, 666
 mappel::QuasiNewtonMaximizer, 717
 mappel::TrustRegionMaximizer, 767
 mcmc.cpp, 805
 mcmc.h, 806
 mcmc_candidate_eta_sigma
 mappel::Gauss2DsxyModel, 503
 mcmc_candidate_eta_y
 mappel::Gauss2DsxyModel, 503
 mean_backtracks
 mappel::IterativeMaximizer, 533
 mappel::NewtonDiagonalMaximizer, 659
 mappel::NewtonMaximizer, 671
 mappel::QuasiNewtonMaximizer, 723
 mappel::TrustRegionMaximizer, 774
 mean_der_evals
 mappel::IterativeMaximizer, 533
 mappel::NewtonDiagonalMaximizer, 659
 mappel::NewtonMaximizer, 671
 mappel::QuasiNewtonMaximizer, 723
 mappel::TrustRegionMaximizer, 774
 mean_fun_evals
 mappel::IterativeMaximizer, 533
 mappel::NewtonDiagonalMaximizer, 659
 mappel::NewtonMaximizer, 671
 mappel::QuasiNewtonMaximizer, 723
 mappel::TrustRegionMaximizer, 774
 mean_iterations
 mappel::IterativeMaximizer, 533
 mappel::NewtonDiagonalMaximizer, 659
 mappel::NewtonMaximizer, 671
 mappel::QuasiNewtonMaximizer, 723
 mappel::TrustRegionMaximizer, 774
 min_scaling
 mappel::TrustRegionMaximizer, 778
 min_sigma
 mappel::Gauss2DsMAP, 389
 mappel::Gauss2DsMLE, 421
 mappel::Gauss2DsModel, 452
 mappel::Gauss2DsxyModel, 503
 model
 mappel::CGaussHeuristicEstimator, 92
 mappel::CGaussMLE, 99
 mappel::Estimator, 105
 mappel::Gauss1DModel::Stencil, 738
 mappel::Gauss1DsModel::Stencil, 742
 mappel::Gauss2DModel::Stencil, 757
 mappel::Gauss2DsModel::Stencil, 752
 mappel::Gauss2DsxyModel::Stencil, 746
 mappel::HeuristicEstimator, 511
 mappel::IterativeMaximizer, 535
 mappel::NewtonDiagonalMaximizer, 661
 mappel::NewtonMaximizer, 674
 mappel::QuasiNewtonMaximizer, 725

- mappel::SimulatedAnnealingMaximizer, 733
 - mappel::ThreadedEstimator, 763
 - mappel::TrustRegionMaximizer, 778
- model_grad
 - PoissonGaussianNoise2DObjective.h, 830
- model_grad2
 - PoissonGaussianNoise2DObjective.h, 830
- model_grad_stack
 - mappel, 39
- model_hessian
 - PoissonGaussianNoise2DObjective.h, 830
- model_hessian_stack
 - mappel, 40
- model_image
 - mappel::methods, 58
 - PoissonGaussianNoise2DObjective.h, 830
- model_image_stack
 - mappel, 40
 - mappel::methods::openmp, 83
- model_methods.h, 814
- model_methods_impl.h, 817
- model_positive_hessian_stack
 - mappel, 41
- ModelBoundsError
 - mappel::ModelBoundsError, 650
- ModelDataStackT
 - mappel, 26
 - mappel::Gauss1DMAP, 111
 - mappel::Gauss1DMLE, 139
 - mappel::Gauss1DsMAP, 194
 - mappel::Gauss1DsMLE, 223
 - mappel::Gauss2DMAP, 278
 - mappel::Gauss2DMLE, 308
 - mappel::Gauss2DsMAP, 367
 - mappel::Gauss2DsMLE, 399
 - mappel::Gauss2DsxyMAP, 461
 - mappel::PoissonNoise1DObjective, 700
 - mappel::PoissonNoise2DObjective, 708
 - PoissonGaussianNoise2DObjective, 696
- ModelDataT
 - mappel, 26
 - mappel::Gauss1DMAP, 112
 - mappel::Gauss1DMLE, 140
 - mappel::Gauss1DsMAP, 195
 - mappel::Gauss1DsMLE, 223
 - mappel::Gauss2DMAP, 278
 - mappel::Gauss2DMLE, 308
 - mappel::Gauss2DsMAP, 367
 - mappel::Gauss2DsMLE, 399
 - mappel::Gauss2DsxyMAP, 462
 - mappel::PoissonNoise1DObjective, 700
 - mappel::PoissonNoise2DObjective, 708
 - PoissonGaussianNoise2DObjective, 696
- modified_cholesky
 - mappel, 41
- mtx
 - mappel::CGaussHeuristicEstimator, 92
 - mappel::CGaussMLE, 99
 - mappel::HeuristicEstimator, 511
 - mappel::IterativeMaximizer, 535
 - mappel::NewtonDiagonalMaximizer, 661
 - mappel::NewtonMaximizer, 674
 - mappel::QuasiNewtonMaximizer, 725
 - mappel::SimulatedAnnealingMaximizer, 733
 - mappel::ThreadedEstimator, 764
 - mappel::TrustRegionMaximizer, 778
- nBacktracks
 - mappel::IterativeMaximizer::MaximizerData, 559
- nIterations
 - mappel::IterativeMaximizer::MaximizerData, 559
- name
 - mappel::CGaussHeuristicEstimator, 91
 - mappel::CGaussMLE, 98
 - mappel::Estimator, 105
 - mappel::Gauss1DMAP, 131
 - mappel::Gauss1DMLE, 159
 - mappel::Gauss1DsMAP, 215
 - mappel::Gauss1DsMLE, 243
 - mappel::Gauss2DMAP, 299
 - mappel::Gauss2DMLE, 329
 - mappel::Gauss2DsMAP, 390
 - mappel::Gauss2DsMLE, 422
 - mappel::Gauss2DsxyMAP, 480
 - mappel::HeuristicEstimator, 510
 - mappel::IterativeMaximizer, 533
 - mappel::NewtonDiagonalMaximizer, 659
 - mappel::NewtonMaximizer, 671
 - mappel::QuasiNewtonMaximizer, 723
 - mappel::SimulatedAnnealingMaximizer, 732
 - mappel::ThreadedEstimator, 763
 - mappel::TrustRegionMaximizer, 774
- negative_definite_hessian
 - mappel::methods::objective, 71
- negative_definite_hessian_stack
 - mappel::methods::objective::openmp, 77
- NewtonDiagonalMaximizer
 - mappel::NewtonDiagonalMaximizer, 653
- NewtonMaximizer
 - mappel::NewtonMaximizer, 666
- normal_prior_grad
 - mappel, 42
- normal_prior_grad2
 - mappel, 42
- normal_quantile_onesided
 - mappel, 42
- normal_quantile_twosided
 - mappel, 42

NotImplementedError
 mappel::NotImplementedError, [676](#)
 num_dim
 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, [390](#)
 mappel::Gauss2DsMLE, [422](#)
 mappel::Gauss2DsModel, [453](#)
 mappel::Gauss2DsxyMap, [480](#)
 mappel::Gauss2DsxyModel, [504](#)
 mappel::ImageFormat1DBase, [517](#)
 mappel::ImageFormat2DBase, [524](#)
 mappel::PoissonNoise1DObjective, [704](#)
 mappel::PoissonNoise2DObjective, [711](#)
 num_estimations
 mappel::CGaussHeuristicEstimator, [92](#)
 mappel::CGaussMLE, [99](#)
 mappel::Estimator, [105](#)
 mappel::HeuristicEstimator, [511](#)
 mappel::IterativeMaximizer, [536](#)
 mappel::NewtonDiagonalMaximizer, [662](#)
 mappel::NewtonMaximizer, [674](#)
 mappel::QuasiNewtonMaximizer, [726](#)
 mappel::SimulatedAnnealingMaximizer, [734](#)
 mappel::ThreadedEstimator, [764](#)
 mappel::TrustRegionMaximizer, [778](#)
 num_hyperparams
 mappel::Gauss1DMap, [132](#)
 mappel::Gauss1DMLE, [160](#)
 mappel::Gauss1DModel, [187](#)
 mappel::Gauss1DsMap, [215](#)
 mappel::Gauss1DsMLE, [243](#)
 mappel::Gauss1DsModel, [270](#)
 mappel::Gauss2DMap, [299](#)
 mappel::Gauss2DMLE, [329](#)
 mappel::Gauss2DModel, [358](#)
 mappel::Gauss2DsMap, [390](#)
 mappel::Gauss2DsMLE, [422](#)
 mappel::Gauss2DsModel, [453](#)
 mappel::Gauss2DsxyMap, [481](#)
 mappel::Gauss2DsxyModel, [504](#)
 mappel::MAPEstimator, [552](#)
 mappel::MCMCAdaptor1Ds, [594](#)
 mappel::MCMCAdaptor1D, [577](#)
 mappel::MCMCAdaptor2Ds, [631](#)
 mappel::MCMCAdaptor2D, [613](#)
 mappel::MLEstimator, [649](#)
 mappel::PointEmitterModel, [694](#)
 num_oversample
 mappel::mcmc, [49](#)
 num_params
 mappel::Gauss1DMap, [132](#)
 mappel::Gauss1DMLE, [160](#)
 mappel::Gauss1DModel, [187](#)
 mappel::Gauss1DsMap, [215](#)
 mappel::Gauss1DsMLE, [243](#)
 mappel::Gauss1DsModel, [270](#)
 mappel::Gauss2DMap, [299](#)
 mappel::Gauss2DMLE, [329](#)
 mappel::Gauss2DModel, [358](#)
 mappel::Gauss2DsMap, [390](#)
 mappel::Gauss2DsMLE, [422](#)
 mappel::Gauss2DsModel, [453](#)
 mappel::Gauss2DsxyMap, [481](#)
 mappel::Gauss2DsxyModel, [504](#)
 mappel::MAPEstimator, [552](#)
 mappel::MCMCAdaptor1Ds, [595](#)
 mappel::MCMCAdaptor1D, [577](#)
 mappel::MCMCAdaptor2Ds, [631](#)
 mappel::MCMCAdaptor2D, [613](#)
 mappel::MLEstimator, [649](#)
 mappel::PointEmitterModel, [694](#)
 num_phases
 mappel::Gauss1DMap, [132](#)
 mappel::Gauss1DMLE, [160](#)
 mappel::Gauss1DModel, [187](#)
 mappel::Gauss1DsMap, [215](#)
 mappel::Gauss1DsMLE, [244](#)
 mappel::Gauss1DsModel, [270](#)
 mappel::Gauss2DMap, [299](#)
 mappel::Gauss2DMLE, [329](#)
 mappel::Gauss2DModel, [358](#)
 mappel::Gauss2DsMap, [390](#)
 mappel::Gauss2DsMLE, [422](#)
 mappel::Gauss2DsModel, [453](#)
 mappel::Gauss2DsxyMap, [481](#)
 mappel::Gauss2DsxyModel, [504](#)
 mappel::MCMCAdaptor1Ds, [595](#)
 mappel::MCMCAdaptor1D, [577](#)
 mappel::MCMCAdaptor2Ds, [631](#)
 mappel::MCMCAdaptor2D, [613](#)
 mappel::MCMCAdaptorBase, [635](#)
 num_threads
 mappel::CGaussHeuristicEstimator, [92](#)
 mappel::CGaussMLE, [99](#)
 mappel::HeuristicEstimator, [511](#)
 mappel::IterativeMaximizer, [536](#)
 mappel::NewtonDiagonalMaximizer, [662](#)
 mappel::NewtonMaximizer, [674](#)
 mappel::QuasiNewtonMaximizer, [726](#)
 mappel::SimulatedAnnealingMaximizer, [734](#)
 mappel::ThreadedEstimator, [764](#)

- mappel::TrustRegionMaximizer, 778
- NumExitCodes
 - mappel::IterativeMaximizer, 536
 - mappel::NewtonDiagonalMaximizer, 662
 - mappel::NewtonMaximizer, 674
 - mappel::QuasiNewtonMaximizer, 726
 - mappel::TrustRegionMaximizer, 779
- numerical.cpp, 820
- numerical.h, 821
- NumericalError
 - mappel::NumericalError, 677
- OMPExcptionCatcher
 - omp_exception_catcher, 85
 - omp_exception_catcher::impl_::OMPExcptionCatcher, 678
- OMPExcptionCatcher.h, 822
- observed_information
 - mappel::methods, 58, 59
- omp_exception_catcher, 85
 - OMPExcptionCatcher, 85
- omp_exception_catcher::impl_, 85
- omp_exception_catcher::impl_::OMPExcptionCatcher
 - Abort, 678
 - Continue, 678
 - DoNotTry, 678
 - OMPExcptionCatcher, 678
 - rethrow, 678
 - RethrowFirst, 678
 - run, 678
 - setGlobalDefaultStrategy, 679
 - Strategy, 678
- omp_exception_catcher::impl_::OMPExcptionCatcher<
 - IntType >, 677
- openmp_methods.h, 823
- operator<<
 - mappel, 42, 43
 - mappel::Estimator, 105
 - mappel::Gauss1DModel::Stencil, 737
 - mappel::Gauss1DsModel::Stencil, 741
 - mappel::Gauss2DModel::Stencil, 756
 - mappel::Gauss2DsModel::Stencil, 750
 - mappel::Gauss2DsxyModel::Stencil, 745
- operator=
 - mappel::Gauss1DMAP, 122
 - mappel::Gauss1DMLE, 150
 - mappel::Gauss1DModel, 177, 178
 - mappel::Gauss1DsMAP, 205
 - mappel::Gauss1DsMLE, 233, 234
 - mappel::Gauss1DsModel, 261
 - mappel::Gauss2DMAP, 290
 - mappel::Gauss2DMLE, 320
 - mappel::Gauss2DModel, 349
 - mappel::Gauss2DsMAP, 379, 380
- mappel::Gauss2DsMLE, 411, 412
- mappel::Gauss2DsModel, 443
- mappel::ImageFormat2DBase, 523
- mappel::MAPEstimator, 546, 547
- mappel::MCMCAdaptor1Ds, 587, 588
- mappel::MCMCAdaptor1D, 570
- mappel::MCMCAdaptor2Ds, 623, 624
- mappel::MCMCAdaptor2D, 605, 606
- mappel::MLEstimator, 644
- mappel::PointEmitterModel, 689
- mappel::PoissonNoise1DObjective, 703
- mappel::PoissonNoise2DObjective, 710
- ParallelRngGeneratorT
 - mappel, 26
- ParallelRngManagerT
 - mappel, 26
- ParamMatT
 - mappel::PriorMAP1DObjective, 713
- ParamVecT
 - mappel, 26
 - mappel::Gauss1DMAP, 112
 - mappel::Gauss1DMLE, 140
 - mappel::Gauss1DModel, 168
 - mappel::Gauss1DsMAP, 195
 - mappel::Gauss1DsMLE, 223
 - mappel::Gauss1DsModel, 251
 - mappel::Gauss2DMAP, 279
 - mappel::Gauss2DMLE, 309
 - mappel::Gauss2DModel, 338
 - mappel::Gauss2DsMAP, 367
 - mappel::Gauss2DsMLE, 399
 - mappel::Gauss2DsModel, 431
 - mappel::Gauss2DsxyMAP, 462
 - mappel::Gauss2DsxyModel, 488
 - mappel::MAPEstimator, 540
 - mappel::MCMCAdaptor1Ds, 581
 - mappel::MCMCAdaptor1D, 563
 - mappel::MCMCAdaptor2Ds, 617
 - mappel::MCMCAdaptor2D, 599
 - mappel::MLEstimator, 638
 - mappel::PointEmitterModel, 683
- ParameterValueError
 - mappel::ParameterValueError, 680
- ParamT
 - mappel, 26
 - mappel::Gauss1DMAP, 112
 - mappel::Gauss1DMLE, 140
 - mappel::Gauss1DModel, 168
 - mappel::Gauss1DModel::Stencil, 735
 - mappel::Gauss1DsMAP, 195
 - mappel::Gauss1DsMLE, 223
 - mappel::Gauss1DsModel, 251
 - mappel::Gauss1DsModel::Stencil, 739

- mappel::Gauss2DModel, 338
- mappel::Gauss2DModel::Stencil, 754
- mappel::Gauss2DsMAP, 367
- mappel::Gauss2DsMLE, 399
- mappel::Gauss2DsModel, 431
- mappel::Gauss2DsModel::Stencil, 748
- mappel::Gauss2DsxyMAP, 462
- mappel::Gauss2DsxyModel, 488
- mappel::Gauss2DsxyModel::Stencil, 743
- mappel::MAPEstimator, 540
- mappel::MCMCAdaptor1Ds, 581
- mappel::MCMCAdaptor1D, 563
- mappel::MCMCAdaptor2Ds, 617
- mappel::MCMCAdaptor2D, 599
- mappel::MLEstimator, 638
- mappel::PointEmitterModel, 683
- mappel::PriorMAP1DObjective, 713
- pareto_prior_grad
 - mappel, 43
- pareto_prior_grad2
 - mappel, 43
- pixel_grad
 - 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, 380
 - mappel::Gauss2DsMLE, 412
 - mappel::Gauss2DsModel, 443
 - mappel::Gauss2DsxyMAP, 472
 - mappel::Gauss2DsxyModel, 497
- pixel_grad2
 - 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, 380
 - mappel::Gauss2DsMLE, 412
 - mappel::Gauss2DsModel, 443
 - mappel::Gauss2DsxyMAP, 472
 - mappel::Gauss2DsxyModel, 497
- pixel_hess
 - 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, 380
 - mappel::Gauss2DsMLE, 412
 - mappel::Gauss2DsModel, 443
 - mappel::Gauss2DsxyMAP, 472
 - mappel::Gauss2DsxyModel, 497
- 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, 380
- mappel::Gauss2DsMLE, 412
- mappel::Gauss2DsModel, 443
- mappel::Gauss2DsxyMAP, 472
- mappel::Gauss2DsxyModel, 498
- pixel_hess_update
 - mappel::Gauss1DModel, 123
 - mappel::Gauss1DModel, 151
 - mappel::Gauss1DModel, 178
 - mappel::Gauss1DsMAP, 206
 - mappel::Gauss1DsMLE, 234
 - mappel::Gauss1DsModel, 262
 - mappel::Gauss2DModel, 291
 - mappel::Gauss2DModel, 321
 - mappel::Gauss2DModel, 350
 - mappel::Gauss2DsMAP, 380
 - mappel::Gauss2DsMLE, 412
 - mappel::Gauss2DsModel, 444
 - mappel::Gauss2DsxyMAP, 472
 - mappel::Gauss2DsxyModel, 498
- pixel_model_value
 - mappel::Gauss1DModel, 123
 - mappel::Gauss1DModel, 151
 - mappel::Gauss1DModel, 178
 - mappel::Gauss1DsMAP, 206
 - mappel::Gauss1DsMLE, 234
 - mappel::Gauss1DsModel, 262
 - mappel::Gauss2DModel, 291
 - mappel::Gauss2DModel, 321
 - mappel::Gauss2DModel, 350
 - mappel::Gauss2DsMAP, 381
 - mappel::Gauss2DsMLE, 413
 - mappel::Gauss2DsModel, 444
 - mappel::Gauss2DsxyMAP, 472
 - mappel::Gauss2DsxyModel, 498
- PointEmitterModel
 - mappel::PointEmitterModel, 683, 684
- PointEmitterModel.cpp, 826
- PointEmitterModel.h, 827
- poisson_log_likelihood
 - mappel, 43
- PoissonGaussianNoise2DObjective
 - CoordIdxT, 696
 - CoordStackT, 696
 - CoordT, 696

- estimator_names, [697](#)
- ModelDataStackT, [696](#)
- ModelDataT, [696](#)
- PoissonGaussianNoise2DObjective, [697](#)
- sensor_bg_map, [697](#)
- sensor_gain_map, [697](#)
- PoissonGaussianNoise2DObjective< ModelBase >, [695](#)
- PoissonGaussianNoise2DObjective.cpp, [828](#)
- PoissonGaussianNoise2DObjective.h, [828](#)
 - fisher_information, [829](#)
 - log_likelihood, [829](#)
 - make_estimator, [829](#)
 - model_grad, [830](#)
 - model_grad2, [830](#)
 - model_hessian, [830](#)
 - model_image, [830](#)
 - relative_log_likelihood, [830](#)
 - simulate_image, [830](#), [831](#)
- PoissonNoise1DObjective
 - mappel::PoissonNoise1DObjective, [701](#)
- PoissonNoise1DObjective.cpp, [831](#)
- PoissonNoise1DObjective.h, [832](#)
- PoissonNoise2DObjective
 - mappel::PoissonNoise2DObjective, [708](#)
- PoissonNoise2DObjective.cpp, [833](#)
- PoissonNoise2DObjective.h, [834](#)
- print_centered_title
 - mappel, [43](#)
- print_image
 - mappel, [44](#)
- print_labeled_image
 - mappel, [44](#)
- print_vec_row
 - mappel, [44](#)
- prior
 - mappel::Gauss1DModel, [132](#)
 - mappel::Gauss1DMLE, [160](#)
 - mappel::Gauss1DModel, [187](#)
 - mappel::Gauss1DsMAP, [216](#)
 - mappel::Gauss1DsMLE, [244](#)
 - mappel::Gauss1DsModel, [271](#)
 - mappel::Gauss2DModel, [300](#)
 - mappel::Gauss2DMLE, [330](#)
 - mappel::Gauss2DModel, [358](#)
 - mappel::Gauss2DsMAP, [391](#)
 - mappel::Gauss2DsMLE, [423](#)
 - mappel::Gauss2DsModel, [453](#)
 - mappel::Gauss2DsxyMAP, [481](#)
 - mappel::Gauss2DsxyModel, [504](#)
 - mappel::MAPEstimator, [552](#)
 - mappel::MCMCAdaptor1Ds, [595](#)
 - mappel::MCMCAdaptor1D, [577](#)
 - mappel::MCMCAdaptor2Ds, [631](#)
 - mappel::MCMCAdaptor2D, [613](#)
 - mappel::MLEstimator, [649](#)
 - mappel::PointEmitterModel, [694](#)
- prior_grad2_update
 - mappel::PriorMAP1DObjective, [713](#)
- prior_grad_update
 - mappel::PriorMAP1DObjective, [713](#)
- prior_hess_update
 - mappel::PriorMAP1DObjective, [713](#)
- prior_log_likelihood
 - mappel::PriorMAP1DObjective, [713](#)
- prior_objective
 - mappel::methods, [59](#)
- prior_relative_log_likelihood
 - mappel::PriorMAP1DObjective, [714](#)
- prior_types
 - mappel::Gauss1DModel, [132](#)
 - mappel::Gauss1DMLE, [160](#)
 - mappel::Gauss1DModel, [187](#)
 - mappel::Gauss1DsMAP, [216](#)
 - mappel::Gauss1DsMLE, [244](#)
 - mappel::Gauss1DsModel, [271](#)
 - mappel::Gauss2DModel, [300](#)
 - mappel::Gauss2DMLE, [330](#)
 - mappel::Gauss2DModel, [359](#)
 - mappel::Gauss2DsMAP, [391](#)
 - mappel::Gauss2DsMLE, [423](#)
 - mappel::Gauss2DsModel, [454](#)
 - mappel::Gauss2DsxyMAP, [481](#)
- PriorMAP1DObjective.h, [836](#)
- psf_sigma
 - mappel::Gauss1DModel, [133](#)
 - mappel::Gauss1DMLE, [161](#)
 - mappel::Gauss1DModel, [188](#)
 - mappel::Gauss2DModel, [300](#)
 - mappel::Gauss2DMLE, [330](#)
 - mappel::Gauss2DModel, [359](#)
 - mappel::Gauss2DsxyMAP, [482](#)
- quadratic_model_value
 - mappel::TrustRegionMaximizer, [774](#)
- QuasiNewtonMaximizer
 - mappel::QuasiNewtonMaximizer, [718](#)
- README.md, [836](#)
- record_backtrack
 - mappel::IterativeMaximizer::MaximizerData, [555](#)
- record_exit
 - mappel::IterativeMaximizer::MaximizerData, [555](#)
- record_iteration
 - mappel::IterativeMaximizer::MaximizerData, [556](#)
- record_run_statistics
 - mappel::IterativeMaximizer, [533](#)
 - mappel::NewtonDiagonalMaximizer, [659](#)
 - mappel::NewtonMaximizer, [671](#)
 - mappel::QuasiNewtonMaximizer, [723](#)

- mappel::TrustRegionMaximizer, 774
- record_walltime
 - mappel::CGaussHeuristicEstimator, 91
 - mappel::CGaussMLE, 98
 - mappel::Estimator, 105
 - mappel::HeuristicEstimator, 510
 - mappel::IterativeMaximizer, 533
 - mappel::NewtonDiagonalMaximizer, 659
 - mappel::NewtonMaximizer, 672
 - mappel::QuasiNewtonMaximizer, 723
 - mappel::SimulatedAnnealingMaximizer, 733
 - mappel::ThreadedEstimator, 763
 - mappel::TrustRegionMaximizer, 775
- refine_gaussian_2Dmax
 - mappel, 44
- refine_gaussian_3Dmax
 - mappel, 44
- reflected_theta
 - mappel::Gauss1DMAP, 123
 - mappel::Gauss1DMLE, 151
 - mappel::Gauss1DModel, 178
 - mappel::Gauss1DsMAP, 206
 - 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, 472
 - mappel::Gauss2DsxyModel, 498
 - mappel::MAPEstimator, 547
 - mappel::MCMCAdaptor1Ds, 588
 - mappel::MCMCAdaptor1D, 570
 - mappel::MCMCAdaptor2Ds, 624
 - mappel::MCMCAdaptor2D, 606
 - mappel::MLEstimator, 644
 - mappel::PointEmitterModel, 689
- reflected_theta_stack
 - 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, 498
- mappel::MAPEstimator, 547
- mappel::MCMCAdaptor1Ds, 588
- mappel::MCMCAdaptor1D, 570
- mappel::MCMCAdaptor2Ds, 624
- mappel::MCMCAdaptor2D, 606
- mappel::MLEstimator, 644
- mappel::PointEmitterModel, 690
- relative_log_likelihood
 - PoissonGaussianNoise2DObjective.h, 830
- relative_poisson_log_likelihood
 - mappel, 45
- rename_hyperparam
 - 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, 445
 - mappel::Gauss2DsxyMAP, 473
 - mappel::Gauss2DsxyModel, 498
 - mappel::MAPEstimator, 547
 - mappel::MCMCAdaptor1Ds, 588
 - mappel::MCMCAdaptor1D, 570
 - mappel::MCMCAdaptor2Ds, 624
 - mappel::MCMCAdaptor2D, 606
 - mappel::MLEstimator, 644
 - mappel::PointEmitterModel, 690
- restore_stencil
 - mappel::IterativeMaximizer::MaximizerData, 556
- restrict_value_range
 - mappel, 45
- rethrow
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, 678
- RethrowFirst
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, 678
- ReturnIfSubclassT
 - mappel, 27
- rho_cauchy_min
 - mappel::TrustRegionMaximizer, 779
- rho_obj_min
 - mappel::TrustRegionMaximizer, 779
- rho_obj_opt
 - mappel::TrustRegionMaximizer, 779
- rlh
 - mappel::IterativeMaximizer::MaximizerData, 559
 - 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, [836](#)
- rng.h, [837](#)
- rng_manager
 - mappel, [47](#)
- RngSeedT
 - mappel, [27](#)
- run
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [678](#)
- s0
 - mappel::IterativeMaximizer::MaximizerData, [559](#)
- s1
 - mappel::IterativeMaximizer::MaximizerData, [559](#)
- sample_mcmc_candidate
 - 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, [445](#)
 - mappel::Gauss2DsxyMAP, [473](#)
 - mappel::Gauss2DsxyModel, [499](#)
 - mappel::MCMCAdaptor1Ds, [588](#)
 - mappel::MCMCAdaptor1D, [571](#)
 - mappel::MCMCAdaptor2Ds, [624](#)
 - mappel::MCMCAdaptor2D, [606](#)
- sample_posterior
 - mappel::mcmc, [49](#)
- sample_posterior_debug
 - mappel::mcmc, [50](#)
- sample_prior
 - 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, [445](#)
 - mappel::Gauss2DsxyMAP, [473](#)
- sample_prior_stack
 - mappel, [45](#)
 - mappel::methods::openmp, [84](#)
- save_seq
 - mappel::IterativeMaximizer::MaximizerData, [559](#)
- save_stencil
 - mappel::IterativeMaximizer::MaximizerData, [556](#)
- saved_stencil
 - mappel::IterativeMaximizer::MaximizerData, [556](#)
- saved_theta
 - mappel::IterativeMaximizer::MaximizerData, [557](#)
- sensor_bg_map
 - PoissonGaussianNoise2DObjective, [697](#)
- sensor_gain_map
 - PoissonGaussianNoise2DObjective, [697](#)
- seq_len
 - mappel::IterativeMaximizer::MaximizerData, [560](#)
- seq_rllh
 - mappel::IterativeMaximizer::MaximizerData, [560](#)
- set_background_mcmc_sampling
 - mappel::Gauss1DMAP, [124](#)
 - mappel::Gauss1DMLE, [152](#)
 - mappel::Gauss1DModel, [179](#)
 - mappel::Gauss1DsMAP, [207](#)
 - 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, [473](#)

- mappel::MCMCAdaptor1Ds, [589](#)
- mappel::MCMCAdaptor1D, [571](#)
- mappel::MCMCAdaptor2Ds, [625](#)
- mappel::MCMCAdaptor2D, [607](#)
- set_bounds
 - 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](#)
 - mappel::Gauss2DsMAP, [382](#)
 - mappel::Gauss2DsMLE, [414](#)
 - mappel::Gauss2DsModel, [445](#)
 - mappel::Gauss2DsxyMAP, [474](#)
 - mappel::Gauss2DsxyModel, [499](#)
 - mappel::MAPEstimator, [547](#)
 - mappel::MCMCAdaptor1Ds, [589](#)
 - mappel::MCMCAdaptor1D, [571](#)
 - mappel::MCMCAdaptor2Ds, [625](#)
 - mappel::MCMCAdaptor2D, [607](#)
 - mappel::MLEstimator, [645](#)
 - mappel::PointEmitterModel, [690](#)
- set_fixed_parameters
 - mappel::IterativeMaximizer::MaximizerData, [557](#)
- set_hyperparam_names
 - 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](#)
 - mappel::Gauss2DsMAP, [382](#)
 - mappel::Gauss2DsMLE, [414](#)
 - mappel::Gauss2DsModel, [446](#)
 - mappel::Gauss2DsxyMAP, [474](#)
 - mappel::Gauss2DsxyModel, [499](#)
 - mappel::MAPEstimator, [548](#)
 - mappel::MCMCAdaptor1Ds, [589](#)
 - mappel::MCMCAdaptor1D, [571](#)
 - mappel::MCMCAdaptor2Ds, [625](#)
 - mappel::MCMCAdaptor2D, [607](#)
 - mappel::MLEstimator, [645](#)
 - mappel::PointEmitterModel, [690](#)
- set_hyperparam_value
 - 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](#)
- mappel::Gauss2DsMAP, [382](#)
- mappel::Gauss2DsMLE, [414](#)
- mappel::Gauss2DsModel, [446](#)
- mappel::Gauss2DsxyMAP, [474](#)
- mappel::Gauss2DsxyModel, [499](#)
- mappel::MAPEstimator, [548](#)
- mappel::MCMCAdaptor1Ds, [589](#)
- mappel::MCMCAdaptor1D, [572](#)
- mappel::MCMCAdaptor2Ds, [625](#)
- mappel::MCMCAdaptor2D, [607](#)
- mappel::MLEstimator, [645](#)
- mappel::PointEmitterModel, [690](#)
- set_hyperparameters
 - mappel::PriorMAP1DObjective, [714](#)
- set_hyperparams
 - 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](#)
 - mappel::Gauss2DsxyModel, [499](#)
 - mappel::MAPEstimator, [548](#)
 - mappel::MCMCAdaptor1Ds, [589](#)
 - mappel::MCMCAdaptor1D, [572](#)
 - mappel::MCMCAdaptor2Ds, [625](#)
 - mappel::MCMCAdaptor2D, [607](#)
 - mappel::MLEstimator, [645](#)
 - mappel::PointEmitterModel, [691](#)
- set_image_in_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, [383](#)
 - mappel::Gauss2DsMLE, [415](#)

- mappel::Gauss2DsModel, [446](#)
- mappel::Gauss2DsxyMAP, [474](#)
- mappel::Gauss2DsxyModel, [500](#)
- mappel::ImageFormat1DBase, [517](#)
- mappel::ImageFormat2DBase, [523](#)
- mappel::PoissonNoise1DObjective, [703](#)
- mappel::PoissonNoise2DObjective, [710](#)
- set_intensity_mcmc_sampling
 - mappel::Gauss1DMAP, [125](#)
 - mappel::Gauss1DMLE, [153](#)
 - mappel::Gauss1DModel, [180](#)
 - 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, [474](#)
 - mappel::MCMCAdaptor1Ds, [589](#)
 - mappel::MCMCAdaptor1D, [572](#)
 - mappel::MCMCAdaptor2Ds, [626](#)
 - mappel::MCMCAdaptor2D, [607](#)
- set_lbound
 - 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, [446](#)
 - mappel::Gauss2DsxyMAP, [475](#)
 - mappel::Gauss2DsxyModel, [500](#)
 - mappel::MAPEstimator, [548](#)
 - mappel::MCMCAdaptor1Ds, [590](#)
 - mappel::MCMCAdaptor1D, [572](#)
 - mappel::MCMCAdaptor2Ds, [626](#)
 - mappel::MCMCAdaptor2D, [608](#)
 - mappel::MLEstimator, [645](#)
 - mappel::PointEmitterModel, [691](#)
- set_max_sigma
 - mappel::Gauss1DsMAP, [209](#)
 - mappel::Gauss1DsMLE, [237](#)
 - mappel::Gauss1DsModel, [264](#)
 - mappel::Gauss2DsMAP, [383](#)
 - mappel::Gauss2DsMLE, [415](#)
 - mappel::Gauss2DsModel, [447](#)
- mappel::Gauss2DsxyModel, [500](#)
- set_max_sigma_ratio
 - mappel::Gauss2DsMAP, [383](#)
 - mappel::Gauss2DsMLE, [415](#)
 - mappel::Gauss2DsModel, [447](#)
 - mappel::Gauss2DsxyModel, [500](#)
- set_mcmc_num_phases
 - mappel::Gauss1DMAP, [126](#)
 - mappel::Gauss1DMLE, [154](#)
 - mappel::Gauss1DModel, [181](#)
 - mappel::Gauss1DsMAP, [209](#)
 - mappel::Gauss1DsMLE, [237](#)
 - mappel::Gauss1DsModel, [265](#)
 - mappel::Gauss2DMAP, [293](#)
 - mappel::Gauss2DMLE, [323](#)
 - mappel::Gauss2DModel, [352](#)
 - mappel::Gauss2DsMAP, [383](#)
 - mappel::Gauss2DsMLE, [415](#)
 - mappel::Gauss2DsModel, [447](#)
 - mappel::Gauss2DsxyMAP, [475](#)
 - mappel::MCMCAdaptor1Ds, [590](#)
 - mappel::MCMCAdaptor1D, [572](#)
 - mappel::MCMCAdaptor2Ds, [626](#)
 - mappel::MCMCAdaptor2D, [608](#)
 - mappel::MCMCAdaptorBase, [634](#)
- set_mcmc_sigma_scale
 - mappel::Gauss1DMAP, [126](#)
 - mappel::Gauss1DMLE, [154](#)
 - mappel::Gauss1DModel, [181](#)
 - mappel::Gauss1DsMAP, [209](#)
 - mappel::Gauss1DsMLE, [237](#)
 - mappel::Gauss1DsModel, [265](#)
 - mappel::Gauss2DMAP, [293](#)
 - mappel::Gauss2DMLE, [323](#)
 - mappel::Gauss2DModel, [352](#)
 - mappel::Gauss2DsMAP, [384](#)
 - mappel::Gauss2DsMLE, [416](#)
 - mappel::Gauss2DsModel, [447](#)
 - mappel::Gauss2DsxyMAP, [475](#)
 - mappel::MCMCAdaptor1Ds, [590](#)
 - mappel::MCMCAdaptor1D, [572](#)
 - mappel::MCMCAdaptor2Ds, [626](#)
 - mappel::MCMCAdaptor2D, [608](#)
 - mappel::MCMCAdaptorBase, [634](#)
- set_min_sigma
 - mappel::Gauss1DsMAP, [209](#), [210](#)
 - mappel::Gauss1DsMLE, [238](#)
 - mappel::Gauss1DsModel, [265](#)
 - mappel::Gauss2DsMAP, [384](#)
 - mappel::Gauss2DsMLE, [416](#)
 - mappel::Gauss2DsModel, [447](#)
 - mappel::Gauss2DsxyModel, [500](#)
- set_model
 - mappel::CGaussHeuristicEstimator, [92](#)

- mappel::CGaussMLE, 98
- mappel::Estimator, 105
- mappel::HeuristicEstimator, 511
- mappel::IterativeMaximizer, 534
- mappel::NewtonDiagonalMaximizer, 660
- mappel::NewtonMaximizer, 672
- mappel::QuasiNewtonMaximizer, 724
- mappel::SimulatedAnnealingMaximizer, 733
- mappel::ThreadedEstimator, 763
- mappel::TrustRegionMaximizer, 775
- set_param_names
 - mappel::Gauss1DMAP, 126
 - mappel::Gauss1DMLE, 154
 - mappel::Gauss1DModel, 181
 - mappel::Gauss1DsMAP, 210
 - mappel::Gauss1DsMLE, 238
 - mappel::Gauss1DsModel, 265
 - mappel::Gauss2DMAP, 293
 - mappel::Gauss2DMLE, 323
 - mappel::Gauss2DModel, 352
 - mappel::Gauss2DsMAP, 384
 - mappel::Gauss2DsMLE, 416
 - mappel::Gauss2DsModel, 447
 - mappel::Gauss2DsxyMAP, 475
 - mappel::Gauss2DsxyModel, 500
 - mappel::MAPEstimator, 548
 - mappel::MCMCAdaptor1Ds, 590
 - mappel::MCMCAdaptor1D, 573
 - mappel::MCMCAdaptor2Ds, 626
 - mappel::MCMCAdaptor2D, 608
 - mappel::MLEstimator, 645
 - mappel::PointEmitterModel, 691
- set_prior
 - mappel::Gauss1DMAP, 126
 - mappel::Gauss1DMLE, 154
 - mappel::Gauss1DModel, 181
 - 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, 448
 - mappel::Gauss2DsxyMAP, 475
 - mappel::Gauss2DsxyModel, 500
 - mappel::MAPEstimator, 548, 549
 - mappel::MCMCAdaptor1Ds, 590
 - mappel::MCMCAdaptor1D, 573
 - mappel::MCMCAdaptor2Ds, 626, 627
 - mappel::MCMCAdaptor2D, 608
 - mappel::MLEstimator, 646
 - mappel::PointEmitterModel, 691
- set_psf_sigma
 - mappel::Gauss1DMAP, 127
 - mappel::Gauss1DMLE, 155
 - mappel::Gauss1DModel, 182
 - mappel::Gauss2DMAP, 294
 - mappel::Gauss2DMLE, 324
 - mappel::Gauss2DModel, 353
 - mappel::Gauss2DsxyMAP, 476
- set_rng_seed
 - mappel::Gauss1DMAP, 127
 - mappel::Gauss1DMLE, 155
 - mappel::Gauss1DModel, 182
 - mappel::Gauss1DsMAP, 210
 - mappel::Gauss1DsMLE, 238
 - mappel::Gauss1DsModel, 266
 - mappel::Gauss2DMAP, 294
 - mappel::Gauss2DMLE, 324
 - mappel::Gauss2DModel, 353
 - mappel::Gauss2DsMAP, 384
 - mappel::Gauss2DsMLE, 416
 - mappel::Gauss2DsModel, 448
 - mappel::Gauss2DsxyMAP, 476
 - mappel::Gauss2DsxyModel, 500
 - mappel::MAPEstimator, 549
 - mappel::MCMCAdaptor1Ds, 591
 - mappel::MCMCAdaptor1D, 573
 - mappel::MCMCAdaptor2Ds, 627
 - mappel::MCMCAdaptor2D, 609
 - mappel::MLEstimator, 646
 - mappel::PointEmitterModel, 691
- set_size
 - mappel::Gauss1DMAP, 127
 - mappel::Gauss1DMLE, 155
 - mappel::Gauss1DModel, 182
 - mappel::Gauss1DsMAP, 210
 - mappel::Gauss1DsMLE, 238, 239
 - mappel::Gauss1DsModel, 266
 - mappel::Gauss2DMAP, 294
 - mappel::Gauss2DMLE, 324
 - mappel::Gauss2DModel, 353
 - mappel::Gauss2DsMAP, 384
 - mappel::Gauss2DsMLE, 416
 - mappel::Gauss2DsModel, 448
 - mappel::Gauss2DsxyMAP, 476
 - mappel::Gauss2DsxyModel, 500
 - mappel::ImageFormat1DBase, 517
 - mappel::ImageFormat2DBase, 524
 - mappel::PoissonNoise1DObjective, 703
 - mappel::PoissonNoise2DObjective, 710
- set_stencil
 - mappel::IterativeMaximizer::MaximizerData, 557
- set_ubound
 - mappel::Gauss1DMAP, 127
 - mappel::Gauss1DMLE, 155

- mappel::Gauss1DModel, [182](#)
- mappel::Gauss1DsMAP, [211](#)
- mappel::Gauss1DsMLE, [239](#)
- mappel::Gauss1DsModel, [266](#)
- mappel::Gauss2DModel, [294](#)
- mappel::Gauss2DMLE, [324](#)
- mappel::Gauss2DModel, [353](#)
- mappel::Gauss2DsMAP, [385](#)
- mappel::Gauss2DsMLE, [417](#)
- mappel::Gauss2DsModel, [448](#)
- mappel::Gauss2DsxyMAP, [476](#)
- mappel::Gauss2DsxyModel, [500](#)
- mappel::MAPEstimator, [549](#)
- mappel::MCMCAdaptor1Ds, [591](#)
- mappel::MCMCAdaptor1D, [573](#)
- mappel::MCMCAdaptor2Ds, [627](#)
- mappel::MCMCAdaptor2D, [609](#)
- mappel::MLEstimator, [646](#)
- mappel::PointEmitterModel, [691](#)
- setGlobalDefaultStrategy
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [679](#)
- sgn
 - mappel, [46](#)
- sigma
 - mappel::Gauss1DsModel::Stencil, [740](#)
- sigma_ratio
 - mappel::Gauss2DsModel::Stencil, [749](#)
- sigma_scale
 - mappel::Gauss1DModel, [133](#)
 - mappel::Gauss1DMLE, [161](#)
 - mappel::Gauss1DModel, [188](#)
 - mappel::Gauss1DsMAP, [216](#)
 - mappel::Gauss1DsMLE, [244](#)
 - mappel::Gauss1DsModel, [271](#)
 - mappel::Gauss2DModel, [300](#)
 - mappel::Gauss2DMLE, [330](#)
 - mappel::Gauss2DModel, [359](#)
 - mappel::Gauss2DsMAP, [391](#)
 - mappel::Gauss2DsMLE, [423](#)
 - mappel::Gauss2DsModel, [454](#)
 - mappel::Gauss2DsxyMAP, [482](#)
 - mappel::MCMCAdaptor1Ds, [595](#)
 - mappel::MCMCAdaptor1D, [577](#)
 - mappel::MCMCAdaptor2Ds, [632](#)
 - mappel::MCMCAdaptor2D, [613](#)
 - mappel::MCMCAdaptorBase, [635](#)
- sigmaX
 - mappel::Gauss2DsModel::Stencil, [749](#)
 - mappel::Gauss2DsxyModel::Stencil, [744](#)
- sigmaY
 - mappel::Gauss2DsModel::Stencil, [750](#)
 - mappel::Gauss2DsxyModel::Stencil, [744](#)
- simulate_image
 - mappel::methods, [59](#), [60](#)
 - PoissonGaussianNoise2DObjective.h, [830](#), [831](#)
- simulate_image_from_model
 - mappel::methods, [60](#), [61](#)
- simulate_image_stack
 - mappel, [46](#)
 - mappel::methods::openmp, [84](#)
- SimulatedAnnealingMaximizer
 - mappel::SimulatedAnnealingMaximizer, [729](#)
- size
 - mappel::Gauss1DModel, [133](#)
 - mappel::Gauss1DMLE, [161](#)
 - mappel::Gauss1DModel, [188](#)
 - mappel::Gauss1DsMAP, [216](#)
 - mappel::Gauss1DsMLE, [245](#)
 - mappel::Gauss1DsModel, [271](#)
 - mappel::Gauss2DModel, [301](#)
 - mappel::Gauss2DMLE, [331](#)
 - mappel::Gauss2DModel, [359](#)
 - mappel::Gauss2DsMAP, [391](#)
 - mappel::Gauss2DsMLE, [423](#)
 - mappel::Gauss2DsModel, [454](#)
 - mappel::Gauss2DsxyMAP, [482](#)
 - mappel::Gauss2DsxyModel, [504](#)
 - mappel::ImageFormat1DBase, [517](#)
 - mappel::ImageFormat2DBase, [524](#)
 - mappel::PoissonNoise1DObjective, [704](#)
 - mappel::PoissonNoise2DObjective, [711](#)
- solve_TR_subproblem
 - mappel::TrustRegionMaximizer, [775](#)
- solve_restricted_step_length_newton
 - mappel::TrustRegionMaximizer, [775](#)
- square
 - mappel, [46](#)
- stackcomp.h, [838](#)
- StatsT
 - mappel, [27](#)
- Stencil
 - mappel::Gauss1DModel::Stencil, [736](#)
 - mappel::Gauss1DsModel::Stencil, [739](#)
 - mappel::Gauss2DModel::Stencil, [754](#)
 - mappel::Gauss2DsModel::Stencil, [748](#)
 - mappel::Gauss2DsxyModel::Stencil, [744](#)
- stencil
 - mappel::IterativeMaximizer::MaximizerData, [557](#)
- stencil.cpp, [839](#)
- stencil.h, [840](#)
- StencilVecT
 - mappel, [27](#)
 - mappel::Gauss1DModel, [112](#)
 - mappel::Gauss1DMLE, [140](#)
 - mappel::Gauss1DModel, [168](#)
 - mappel::Gauss1DsMAP, [195](#)
 - mappel::Gauss1DsMLE, [223](#)

- mappel::Gauss1DsModel, [251](#)
- mappel::Gauss2DMap, [279](#)
- mappel::Gauss2DMLE, [309](#)
- mappel::Gauss2DModel, [338](#)
- mappel::Gauss2DsMap, [367](#)
- mappel::Gauss2DsMLE, [399](#)
- mappel::Gauss2DsModel, [431](#)
- mappel::Gauss2DsxyMap, [462](#)
- mappel::Gauss2DsxyModel, [488](#)
- StencilT
 - mappel, [27](#)
- step
 - mappel::IterativeMaximizer::MaximizerData, [560](#)
- StepSize
 - mappel::IterativeMaximizer, [528](#)
 - mappel::NewtonDiagonalMaximizer, [653](#)
 - mappel::NewtonMaximizer, [666](#)
 - mappel::QuasiNewtonMaximizer, [717](#)
 - mappel::TrustRegionMaximizer, [768](#)
- Strategy
 - omp_exception_catcher::impl_::OMPEXception←
Catcher, [678](#)
- StringVecT
 - mappel, [27](#)
- T_init
 - mappel::SimulatedAnnealingMaximizer, [734](#)
- 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, [738](#)
 - mappel::Gauss1DsModel::Stencil, [742](#)
 - mappel::Gauss2DModel::Stencil, [757](#)
 - mappel::Gauss2DsModel::Stencil, [752](#)
 - mappel::Gauss2DsxyModel::Stencil, [747](#)
 - mappel::IterativeMaximizer::MaximizerData, [557](#)
- theta_in_bounds
 - mappel::Gauss1DMap, [127](#)
 - mappel::Gauss1DMLE, [155](#)
 - mappel::Gauss1DModel, [182](#)
 - mappel::Gauss1DsMap, [211](#)
 - mappel::Gauss1DsMLE, [239](#)
 - mappel::Gauss1DsModel, [266](#)
 - mappel::Gauss2DMap, [294](#)
 - mappel::Gauss2DMLE, [324](#)
 - mappel::Gauss2DModel, [353](#)
 - mappel::Gauss2DsMap, [385](#)
 - mappel::Gauss2DsMLE, [417](#)
 - mappel::Gauss2DsModel, [448](#)
 - mappel::Gauss2DsxyMap, [476](#)
 - mappel::Gauss2DsxyModel, [500](#)
 - mappel::MAPEstimator, [549](#)
 - mappel::MCMCAdaptor1Ds, [591](#)
 - mappel::MCMCAdaptor1D, [573](#)
 - mappel::MCMCAdaptor2Ds, [627](#)
 - mappel::MCMCAdaptor2D, [609](#)
 - mappel::MLEstimator, [646](#)
 - mappel::PointEmitterModel, [692](#)
- theta_seq
 - mappel::IterativeMaximizer::MaximizerData, [560](#)
- theta_stack_in_bounds
 - 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, [501](#)
 - mappel::MAPEstimator, [549](#)
 - mappel::MCMCAdaptor1Ds, [591](#)
 - mappel::MCMCAdaptor1D, [574](#)

- mappel::MCMCAdaptor2Ds, [627](#)
 - mappel::MCMCAdaptor2D, [609](#)
 - mappel::MLEstimator, [646](#)
 - mappel::PointEmitterModel, [692](#)
- thin_sample
 - mappel::mcmc, [50](#)
- ThreadedEstimator
 - mappel::ThreadedEstimator, [759](#)
- total_backtracks
 - mappel::IterativeMaximizer, [536](#)
 - mappel::NewtonDiagonalMaximizer, [662](#)
 - mappel::NewtonMaximizer, [674](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::TrustRegionMaximizer, [779](#)
- total_der_evals
 - mappel::IterativeMaximizer, [536](#)
 - mappel::NewtonDiagonalMaximizer, [662](#)
 - mappel::NewtonMaximizer, [675](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::TrustRegionMaximizer, [779](#)
- total_fun_evals
 - mappel::IterativeMaximizer, [536](#)
 - mappel::NewtonDiagonalMaximizer, [662](#)
 - mappel::NewtonMaximizer, [675](#)
 - mappel::QuasiNewtonMaximizer, [726](#)
 - mappel::TrustRegionMaximizer, [779](#)
- total_iterations
 - mappel::IterativeMaximizer, [536](#)
 - mappel::NewtonDiagonalMaximizer, [663](#)
 - mappel::NewtonMaximizer, [675](#)
 - mappel::QuasiNewtonMaximizer, [727](#)
 - mappel::TrustRegionMaximizer, [779](#)
- total_walltime
 - mappel::CGaussHeuristicEstimator, [93](#)
 - mappel::CGaussMLE, [99](#)
 - mappel::Estimator, [106](#)
 - mappel::HeuristicEstimator, [512](#)
 - mappel::IterativeMaximizer, [537](#)
 - mappel::NewtonDiagonalMaximizer, [663](#)
 - mappel::NewtonMaximizer, [675](#)
 - mappel::QuasiNewtonMaximizer, [727](#)
 - mappel::SimulatedAnnealingMaximizer, [734](#)
 - mappel::ThreadedEstimator, [764](#)
 - mappel::TrustRegionMaximizer, [780](#)
- TrustRegionMaximizer
 - mappel::TrustRegionMaximizer, [768](#)
- TrustRegionRadius
 - mappel::IterativeMaximizer, [528](#)
 - mappel::NewtonDiagonalMaximizer, [653](#)
 - mappel::NewtonMaximizer, [666](#)
 - mappel::QuasiNewtonMaximizer, [717](#)
 - mappel::TrustRegionMaximizer, [768](#)
- ubound
 - mappel::Gauss1DModel, [133](#)
 - mappel::Gauss1DModel, [161](#)
 - mappel::Gauss1DModel, [188](#)
 - mappel::Gauss1DsMAP, [217](#)
 - mappel::Gauss1DsMLE, [245](#)
 - mappel::Gauss1DsModel, [272](#)
 - mappel::Gauss2DModel, [301](#)
 - mappel::Gauss2DModel, [331](#)
 - mappel::Gauss2DModel, [360](#)
 - mappel::Gauss2DsMAP, [392](#)
 - mappel::Gauss2DsMLE, [424](#)
 - mappel::Gauss2DsModel, [455](#)
 - mappel::Gauss2DsxyMAP, [482](#)
 - mappel::Gauss2DsxyModel, [505](#)
 - mappel::IterativeMaximizer::MaximizerData, [560](#)
 - mappel::MAPEstimator, [552](#)
 - mappel::MCMCAdaptor1Ds, [596](#)
 - mappel::MCMCAdaptor1D, [578](#)
 - mappel::MCMCAdaptor2Ds, [632](#)
 - mappel::MCMCAdaptor2D, [614](#)
 - mappel::MLEstimator, [649](#)
 - mappel::PointEmitterModel, [694](#)
- Unassigned
 - mappel::IterativeMaximizer, [528](#)
 - mappel::NewtonDiagonalMaximizer, [653](#)
 - mappel::NewtonMaximizer, [666](#)
 - mappel::QuasiNewtonMaximizer, [717](#)
 - mappel::TrustRegionMaximizer, [768](#)
- UniformDistT
 - mappel, [27](#)
- update_internal_1D_estimators
 - mappel::Gauss2DsxyModel, [501](#)
- update_internal_1Dsum_estimators
 - mappel::Gauss2DModel, [295](#)
 - mappel::Gauss2DModel, [325](#)
 - mappel::Gauss2DModel, [354](#)
 - mappel::Gauss2DsMAP, [385](#)
 - mappel::Gauss2DsMLE, [417](#)
 - mappel::Gauss2DsModel, [449](#)
- util.cpp, [842](#)
- util.h, [842](#)
- VecFieldT
 - mappel, [27](#)
- VecT
 - mappel, [27](#)
- X
 - mappel::Gauss1DModel::Stencil, [738](#)
 - mappel::Gauss1DsModel::Stencil, [742](#)
 - mappel::Gauss2DModel::Stencil, [757](#)
 - mappel::Gauss2DsModel::Stencil, [752](#)
 - mappel::Gauss2DsxyModel::Stencil, [747](#)
- x
 - mappel::Gauss1DModel::Stencil, [736](#)

mappel::Gauss1DsModel::Stencil, [740](#)
mappel::Gauss2DModel::Stencil, [755](#)
mappel::Gauss2DsModel::Stencil, [750](#)
mappel::Gauss2DsxyModel::Stencil, [744](#)

x_model

mappel::Gauss2DMAP, [301](#)
mappel::Gauss2DMLE, [331](#)
mappel::Gauss2DModel, [360](#)
mappel::Gauss2DsMAP, [392](#)
mappel::Gauss2DsMLE, [424](#)
mappel::Gauss2DsModel, [455](#)
mappel::Gauss2DsxyModel, [505](#)

Y

mappel::Gauss2DModel::Stencil, [757](#)
mappel::Gauss2DsModel::Stencil, [753](#)
mappel::Gauss2DsxyModel::Stencil, [747](#)

y

mappel::Gauss2DModel::Stencil, [755](#)
mappel::Gauss2DsModel::Stencil, [750](#)
mappel::Gauss2DsxyModel::Stencil, [745](#)

y_model

mappel::Gauss2DMAP, [302](#)
mappel::Gauss2DMLE, [332](#)
mappel::Gauss2DModel, [360](#)
mappel::Gauss2DsMAP, [392](#)
mappel::Gauss2DsMLE, [424](#)
mappel::Gauss2DsModel, [455](#)
mappel::Gauss2DsxyModel, [505](#)