The QUESO Library

User's Manual
Version 0.41.0
October 30, 2009

Quantification of Uncertainty for Estimation, Simulation, and Optimization (QUESO)

Contributors:
Paul T. Bauman
Sai Hung Cheung
Todd A. Oliver
Ernesto E. Prudencio
Karl W. Schulz
Rhys Ulerich

Center for Predictive Engineering and Computational Sciences (PECOS)
Institute for Computational and Engineering Sciences (ICES)
The University of Texas at Austin

Abstract

QUESO is a collection of algorithms and C++ classes aimed for research in uncertainty quantification, including the solution of statistical inverse and statistical forward problems, the validation of mathematical models under uncertainty and the prediction of quantities of interest from such models along with the quantification of their uncertainties.

QUESO is designed for flexibility, portability, easiness of use and easiness of extension. Its software design follows an object-oriented approach and its code is written on C++ and over MPI. It can run over uniprocessor or multiprocessor environments.

QUESO contains two forms of documentation: a User's Manual available in pdf format and a lower-level code documentation available in web based/html format.

This is the User's Manual. It gives an overview of the QUESO capabilities, provides procedures for software execution, and includes example studies.

Disclaimer (To be checked by Karl)

Contents

Preface

The QUESO project started in 2008 as part of the efforts of the recently established Center for Predictive Engineering and Computational Sciences (PECOS) at the Institute for Computational and Engineering Sciences (ICES) at The University of Texas at Austin.

The PECOS Center was selected by the National Nuclear Security Administration (NNSA) as one of its new five centers of excellence under the Predictive Science Academic Alliance Program (PSAAP). The goal of the PECOS Center is to advance predictive science and to develop the next generation of advanced computational methods and tools for the calculation of reliable predictions on the behavior of complex phenomena and systems (multiscale, multidisciplinary). This objective demands a systematic, comprehensive treatment of the calibration and validation of the mathematical models involved, as well as the quantification of the uncertainties inherent in such models. The advancement of predictive science is essential for the application of Computational Science to the solution of realistic problems of national interest.

The QUESO library, since its first version, has been publicly released as open source under the GNU General Public License and is available for free download world-wide. See http://www.gnu.org/licenses/gpl.html for more information on the GPL software use agreement.

The QUESO development team currently consists of Paul T. Bauman, Sai Hung Cheung, Todd A. Oliver, Ernesto E. Prudencio, Karl W. Schulz, and Rhys Ulerich.

Contact Information:
Ernesto E. Prudencio
Institute for Computational and Engineering Sciences
1 University Station C0200
Austin, Texas 78712

email: prudenci@ices.utexas.edu
web: http://pecos.ices.utexas.edu

Referencing the QUESO Library

When referencing the QUESO library in a publication, please cite the following:

```
@Misc{queso-web-page,
   Author = "Ernesto E. Prudencio and Paul T. Bauman and Sai Hung Cheung
             and Todd A. Oliver and Karl W. Schulz and Rhys Ulerich",
   Title  = "{T}he {QUESO} {L}ibrary: {Q}uantification of {U}ncertainty
             for {E}stimation, {S}imulation and {O}ptimization",
```

```
   Note   = "http://pecos.ices.utexas.edu",
   Year   = "2008-2009"}

@TechReport{queso-user-ref,
   Author      = "Ernesto E. Prudencio and Paul T. Bauman and Sai Hung Cheung
                  and Todd A. Oliver and Karl W. Schulz and Rhys Ulerich",
   Title       = "{T}he {QUESO} {L}ibrary, {U}ser's {M}anual, {ICES} {T}echnical
                  {R}eport XXYYZZ",
   Institution = "Center for Predictive Engineering and Computational Sciences
                  (PECOS), at the Institute for Computational and Engineering
                  Sciences (ICES), The University of Texas at Austin",
   Year        = "2008-2009"}
```


Acknowledgments

Portions of this material is based upon work supported by the Department of Energy [National Nuclear Security Administration] under Award Number [DE-FC52-08NA28615].

We would also like to thank James Martin, Roy Stogner and Lucas Wilcox for interesting discussions and constructive feedbacks.

Chapter 1

Introduction (Incomplete)

The QUESO library is able to handle uni- and multi-processor Linux environments.

1.1 Key Statistical Concepts

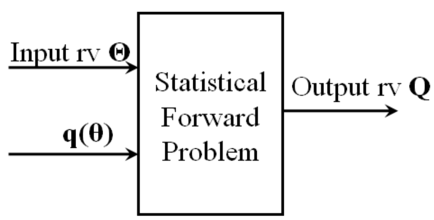

Figure 1.1.1: The representation of a statistical forward problem. $\mathbf{\Theta}$ denotes a random variable related to parameters, $\boldsymbol{\theta}$ denotes a realization of $\mathbf{\Theta}$ and $\mathbf{Q}$ denotes a random variable related to quantities of interest.

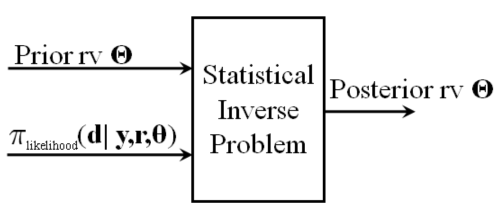

Figure 1.1.2: The representation of a statistical inverse problem. $\mathbf{\Theta}$ denotes a random variable related to parameters, $\boldsymbol{\theta}$ denotes a realization of $\mathbf{\Theta}$ and $\mathbf{r}$ denotes model equations, $\mathbf{y}$ denotes some model output data and $\mathbf{d}$ denotes experimental data.

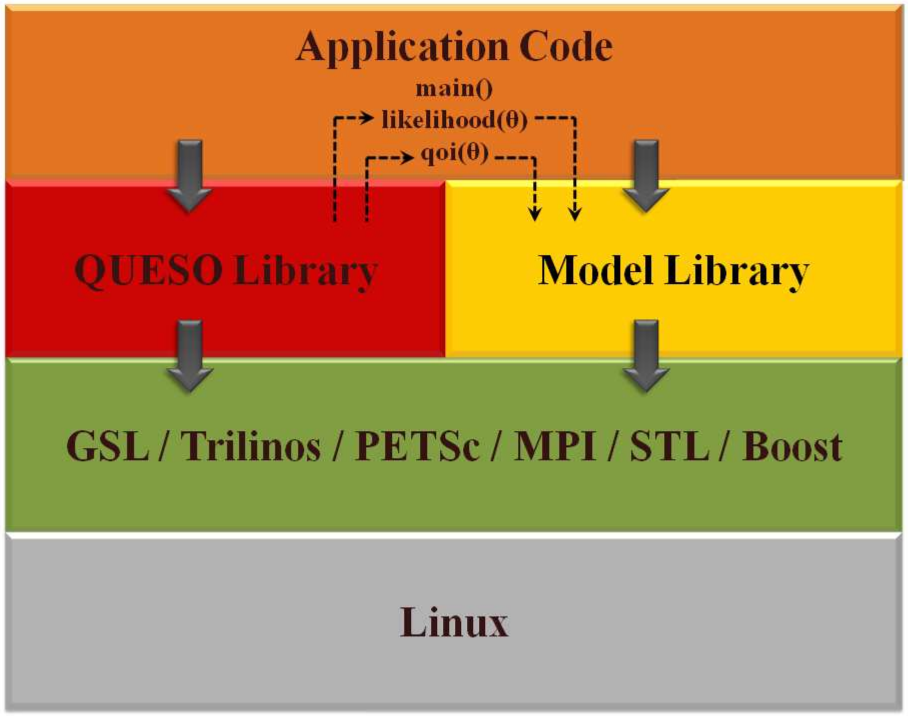

Figure 1.2.1: Overview of the software stack of a typical application that uses QUESO. The symbol $\boldsymbol{\theta}$ represents a vector of $n \geqslant 1$ parameters. Algorithms in the QUESO library require the supply of a likelihood routine $\pi_{\text{like}} : \mathbb{R}^n \rightarrow \mathbb{R}_+$ for statistical inverse problems and of a qoi routine $\mathbf{q} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ for statistical forward problems. These routines exist at the application level and provide the necessary bridge between the statistical algorithms in QUESO, model knowledge in the model library and scenario and experimental data in the disk space. Concepts are further detailed in Chapter 1.

1.2 The Software Stack of an Application Using QUESO

1.2.1 A QUESO Environment

1.2.2 Using Other C++ Classes in the Library

1.2.3 Input and Output Files

Chapter 2

Installation

This chapter describes how to install QUESO, test it and use it to create your application.

2.1 Installation Steps

There are eight steps to make the QUESO Library available at your LINUX computing system. They are listed below, with examples of commands:

1. prepare your LINUX environment (assuming csh; some commands might be enough):
   - module load gnu
   - module load openmpi
   - `setenv LD_LIBRARY_PATH \$LD_LIBRARY_PATH:`
     `/home/johndoe/Installations/gsl_1_12/lib:`
     `/home/johndoe/Installations/boost_1_37_0/lib:`
     `/home/johndoe/Installations/hpct_0_25_1/lib`
   - setenv CC gcc
   - setenv CXX g++
   - setenv MPICC mpicc
   - setenv MPICXX mpic++
   - setenv F77 f77
   - setenv FC gfortran
2. install five packages:
   - GNU Scientific Library (GSL) [2], e.g GSL 1.12,
   - Boost C++ Libraries [1], e.g. Boost 1.37.0,
   - MPI Library, e.g. Open MPI [5] or MPICH [4],
   - Trilinos Library [3], e.g. Trilinos 9.0.2, and
   - High Performance Computing Toolkit (HPCT) [6], e.g. HPCT 0.25.1.

  - ./configure --prefix=/home/johndoe/Installations/hpct_0_25_1 \
    --with-boost=/home/johndoe/Installations/boost_1_37_0
  - make
  - make install
  - note: the directory '/home/johndoe/Installations/hpct_0_25_1' does not need to exist in advance, since it will be created by the command 'make install' above.

3. untar the QUESO tar.gz file (more comments in Section 2.2):
   - cd /home/johndoe
   - mkdir queso_download
   - cd /home/johndoe/queso_download
   - mv <ORIGINAL_LOCATION>queso-0.41.0.tar.gz .
   - tar -zxvf queso-0.41.0.tar
4. configure the QUESO building environment:
   - cd /home/johndoe/queso_download/queso-0.41.0
   - ./configure --prefix=/home/johndoe/Installations/queso_0_41_0_gnu \
     --with-trilinos=/home/johndoe/Installations/trilinos_9_0_2 \
     --with-boost=/home/johndoe/Installations/boost_1_37_0 \
     --with-gsl-prefix=/home/johndoe/Installations/gsl_1_12 \
     --with-hpct-prefix=/home/johndoe/Installations/hpct_0_25_1 \
     CXXFLAGS=''-DMPICH_IGNORE_CXX_SEEK -O3 -Wall -wd383 -wd981 -wd1572''
   - if you want to see the full list of configure options, just run "./configure –help"
   - note: the directory '/home/johndoe/Installations/queso_0_41_0_gnu' does not need to exist in advance, since it will be created in step 7.
5. compile the QUESO source code (library, examples and tests):
   - make
6. check the compiled source (more comments in Section 2.3):
   - make check
7. install the QUESO library (more comments in Section 2.5):
   - make install
8. create the documentation in html format:
   - make docs
   - firefox docs/html/index.html

2.2 The Source Directory Structure

The QUESO source directory contains three main directories. They are listed below and more information about them can be obtained with the html documentation from step 8 above:

- ’libs’, with five subdirectories:
  - ’libs/core/’, with ’inc’ and ’src’ subdirectories,
  - ’libs/misc/’, with ’inc’ and ’src’ subdirectories,
  - ’libs/basic/’, with ’inc’ and ’src’ subdirectories,
  - ’libs/stats/’, with ’inc’ and ’src’ subdirectories, and
  - ’libs/interface/’.
- ’examples’, with four subdirectories:
  - ’examples/statisticalForwardProblem/’,
  - ’examples/statisticalInverseProblem1/’,
  - ’examples/validationCycle/’, and
  - ’examples/validationCycle2/’.
- ’test’, with four subdirectories:
  - ’test/t01_valid_cycle/’,
  - ’test/t02_sip_sfp/’,
  - ’test/t03_sequence/’, and
  - ’text/gsl_tests’.

The executables under ’examples/validationCycle2/’, ’test/t02_sip_sfp/’, ’test/t03_sequence/’ and ’test/gsl_tests/’ have the majority of their codes in *.C files. They might then be easier to understand than the other exectuables in ’examples’ and ’test/t01_valid_cycle’, which have the majority of their codes in *.h files, with templated routines. It should be clear, though, that all executables might be implemented in either *.h or *.C files. It is a matter of how generic you want your application to be.

2.3 Checking the Compiled Source

Just run ’make ckeck’ at the same directory where ’configure’ and ’make’ were run. Many printouts will appear in the screen, but towards the end of them you should see a message like:

```
==================
All 2 tests passed
==================
```

The 2 tests mentioned in this message are the ones under 'test/t01_valid_cycle' and 'test/t02_sip_sfp'. These tests are used as part of the periodic QUESO regression tests. The code for 't02_sip_sfp' is mentioned in Subsection 2.4.3 and is explained in more detail in Chapter 4.

2.4 Running the Executables Provided with QUESO

This section assumes that you have successfully executed steps 1 through 6 above. The codes listed in this section have explanations inside themselves, and some of them print messages during execution to make it clearer what is going on.

2.4.1 Executable at 'examples/statisticalInverseProblem1/'

Just run the following commands:

- cd /home/johndoe/queso_download/queso-0.41.0/
- cd examples/statisticalInverseProblem1/tests/test_2009_02_03/
- rm outputData/*
- ../../src/exStatisticalInverseProblem1_gsl sip.inp [this will take some seconds]
- matlab
- [inside matlab] sip_plot
- [press the left button of the mouse at a picture displayed by 'sip_plot.m', in order to display the next picture]
- [inside matlab] exit
- ls -l outputData/*.png

2.4.2 Executable at 'examples/statisticalForwardProblem/'

Just run the following commands:

- cd /home/johndoe/queso_download/queso-0.41.0/
- cd examples/statisticalForwardProblem1/tests/test_2009_02_11/
- rm outputData/*
- ../../src/exStatisticalForwardProblem1_gsl sfp.inp [this will take some seconds]
- matlab
- [inside matlab] sfp_plot
- [press the left button of the mouse at a picture displayed by 'sfp_plot.m', in order to display the next picture]
- [inside matlab] exit
- ls -l outputData/*.png

2.4.3 Executable at 'test/t02_sip_sfp/sip_sfp/'

Just run the following commands:

- cd /home/johndoe/queso_download/queso-0.41.0/
- cd test/t02_sip_sfp/sip_sfp/
- rm outputData/*
- ./SipSfpExample_gsl example.inp [this will take some seconds]
- matlab
- [inside matlab] example_plots
- [press the left button of the mouse at a picture displayed by 'example_plots.m', in order to display the next picture]
- [inside matlab] exit
- ls -l outputData/*.png

2.5 The Installed Directory Structure

This section assumes you have successfully executed steps 1 through 7 above. The QUESO installed directory contains three main directories:

- 'lib',
- 'include', and
- 'examples', with two subdirectories:
  - 'examples/basic/',
  - 'examples/advanced/'.

2.6 Create your Application with the installed QUESO

Prepare your environment by running

```
setenv LD_LIBRARY_PATH \$LD_LIBRARY_PATH:
       /home/johndoe/Installations/queso_0_41_0_gnu/lib
```

An example Makefile is given below:

```
# BEGIN OF MAKEFILE
QUESO_DIR = /home/johndoe/Installations/queso_0_41_0_gnu/
TRILINOS_DIR = /home/johndoe/Installations/trilinos_9_0_2/
BOOST_DIR = /home/johndoe/Installations/boost_1_37_0/
GSL_DIR = /home/johndoe/Installations/gsl_1_12/
HPCT_DIR = /home/johndoe/Installations/hpct_0_25_1/

include $(TRILINOS_DIR)/include/Makefile.export.epetra

INC_PATHS = \
-I. \
-I$(QUESO_DIR)/include \
-I$(MPI_DIR)/include \
-I$(BOOST_DIR)/include/boost_1_37_0 \
-I$(GSL_DIR)/include \
-I$(HPCT_DIR)/include \
```

```
$(EPETRA_INCLUDES)

LIBS = \
-L$(QUESO_DIR)/lib \
-lqueso \
-L$(MPI_DIR)/lib \
-L$(TRILINOS_DIR)/lib \
-L$(BOOST_DIR)/lib \
-lboost_program_options \
-L$(GSL_DIR)/lib \
-lgsl \
-L$(HPCT_DIR)/lib \
-lhpct \
$(EPETRA_LIBS)

CXX = mpic++
CXXFLAGS += -O3 -Wall -c

default: all

.SUFFIXES: .o .C

all: ex_gsl

clean:
rm -f *~
rm -f *.o
rm -f example

ex_gsl: example_main.o example_likelihood.o example_qoi.o example_compute.o
$(CXX) example_main.o \
       example_likelihood.o \
       example_qoi.o \
       example_compute.o \
       -o example_gsl $(LIBS)

%.o: %.C
$(CXX) $(INC_PATHS) $(CXXFLAGS) $<
# END OF MAKEFILE
```

More documentation is provided in Chapter 4.

Chapter 3

C++ Classes in the Library (Incomplete)

3.1 Core Classes

There are four core classes:

- environment and environment options (Figures 3.1.1 and 3.1.2, pages 12 and 13),
- vector (Figure 3.1.3, page 14),
- matrix (Figure 3.1.4, page 15).

3.1.1 Environment (and Options)

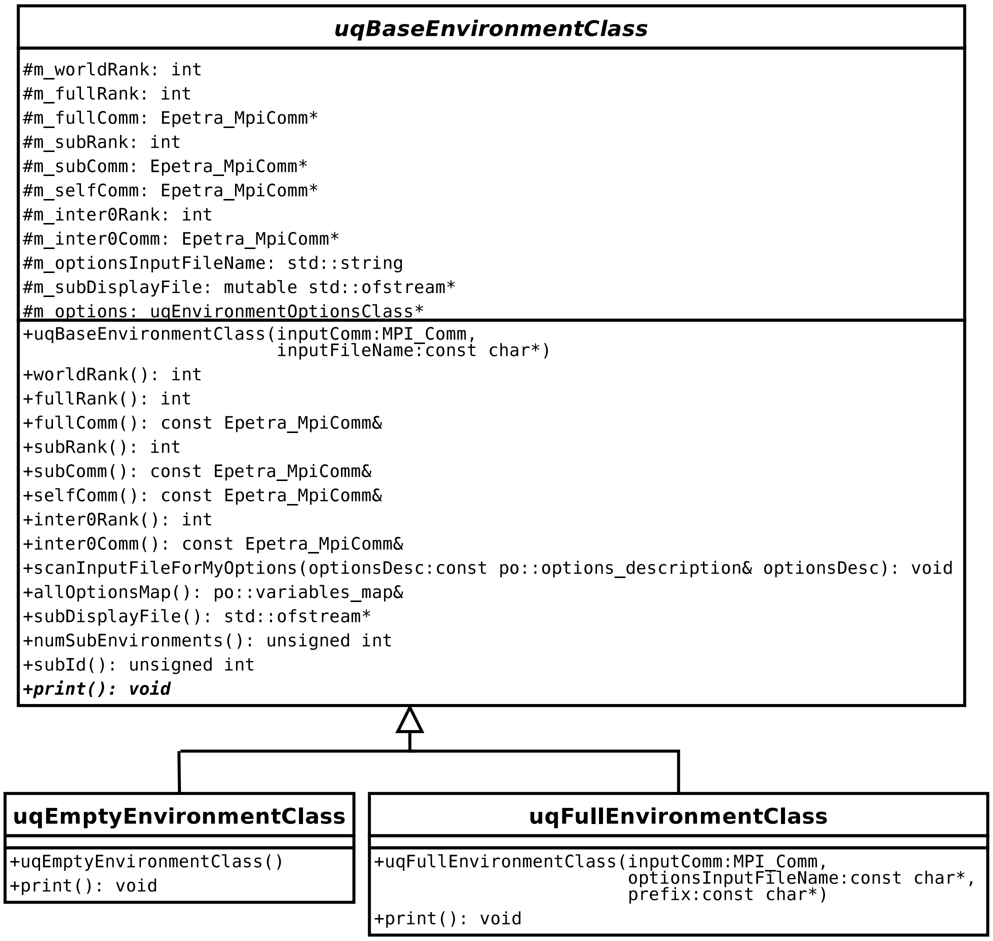

Figure 3.1.1: The class diagram for the environment class.

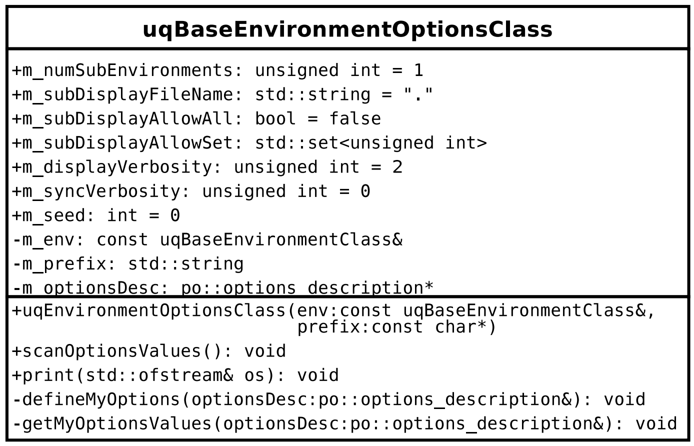

Figure 3.1.2: The environment options class.

Option Name	Default Value	Description
⟨PREFIX⟩env_help		
⟨PREFIX⟩env_numSubEnvironments		
⟨PREFIX⟩env_subDisplayFileName		
⟨PREFIX⟩env_subDisplayAllowAll		
⟨PREFIX⟩env_subDisplayAllowedSet		
⟨PREFIX⟩env_displayVerbosity		
⟨PREFIX⟩env_syncVerbosity		
⟨PREFIX⟩env_seed		

Table 3.1.1: Input file options for a QUESO environment.

3.1.2 Vector

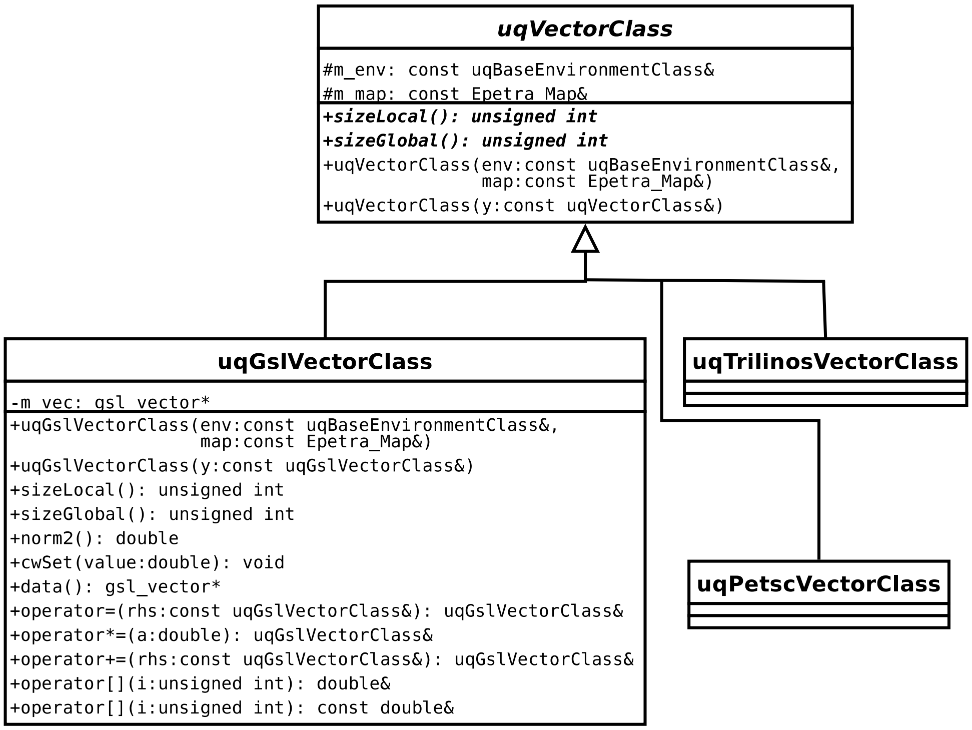

Figure 3.1.3: The class diagram for the vector class.

3.1.3 Matrix

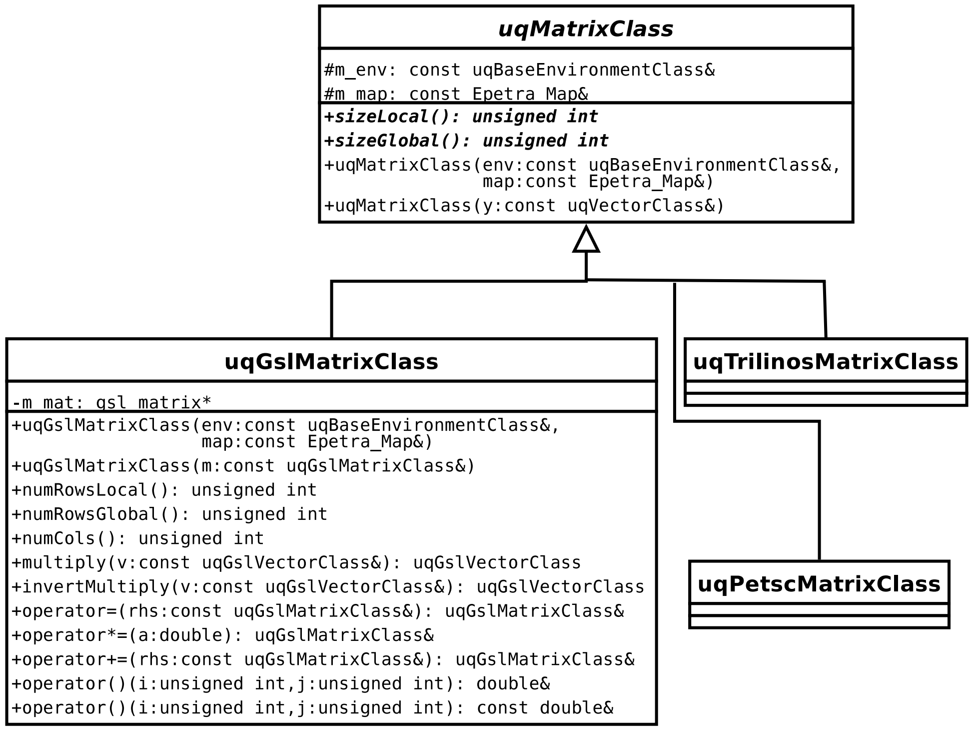

Figure 3.1.4: The class diagram for the matrix class.

3.2 Miscellaneous Classes and Routines

3.3 Templated Basic Classes

The classes in this group are:

- Vector sets, subsets and spaces (see Figure 3.3.1),
- Scalar function (see Figure 3.3.2),
- Vector function (see Figure 3.3.3),
- Scalar sequence (see Figure 3.3.4), and
- Vector sequence (see Figure 3.3.5).

These classes constitute the core entities necessary for the formal mathematical definition and description of other entities, such as random variables, Bayesian solutions of inverse problems, sampling algorithms and chains.

3.3.1 Vector Subset and Vector Space

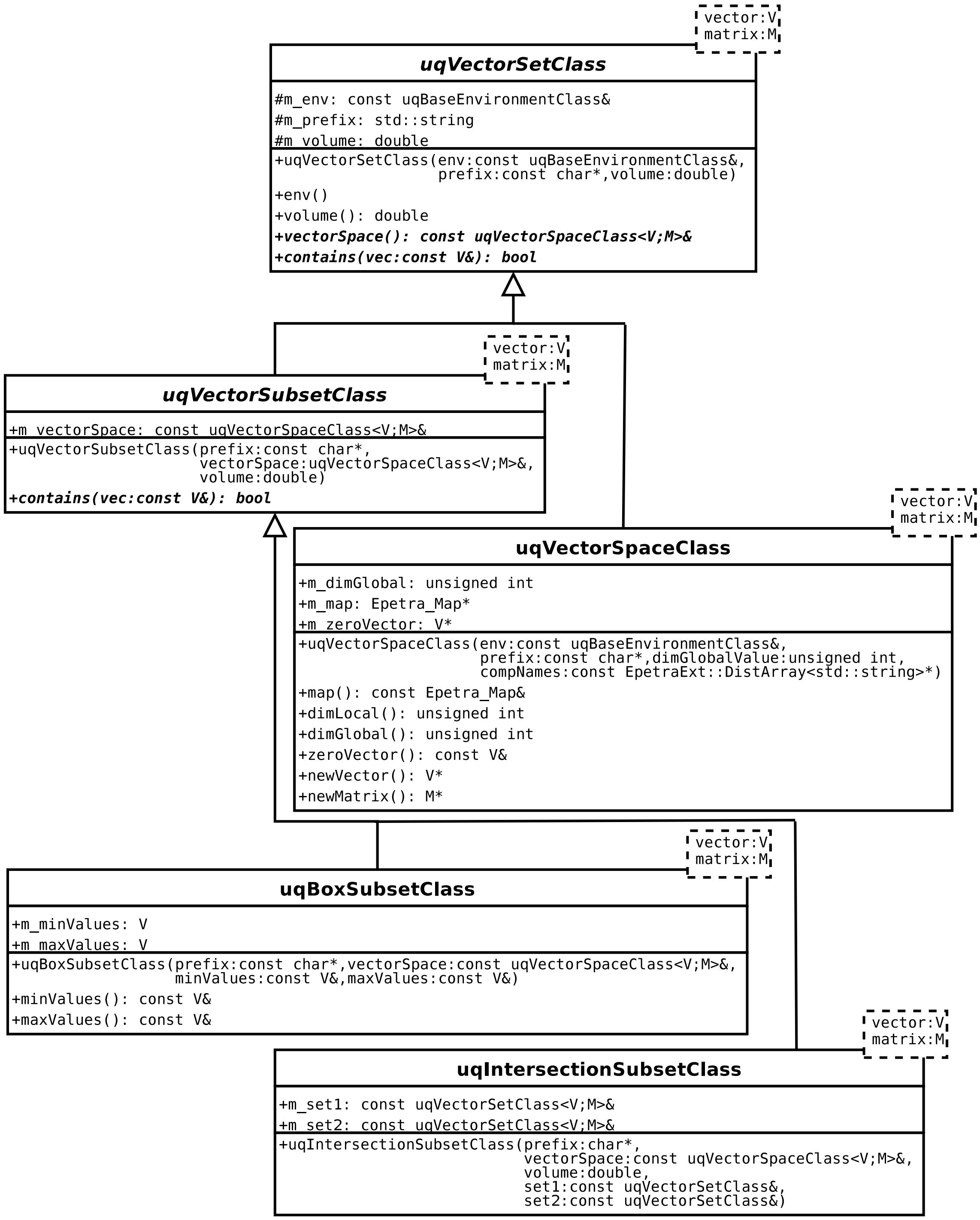

Figure 3.3.1: The class diagram for vector set, vector subset and vector space classes.

3.3.2 Scalar Function

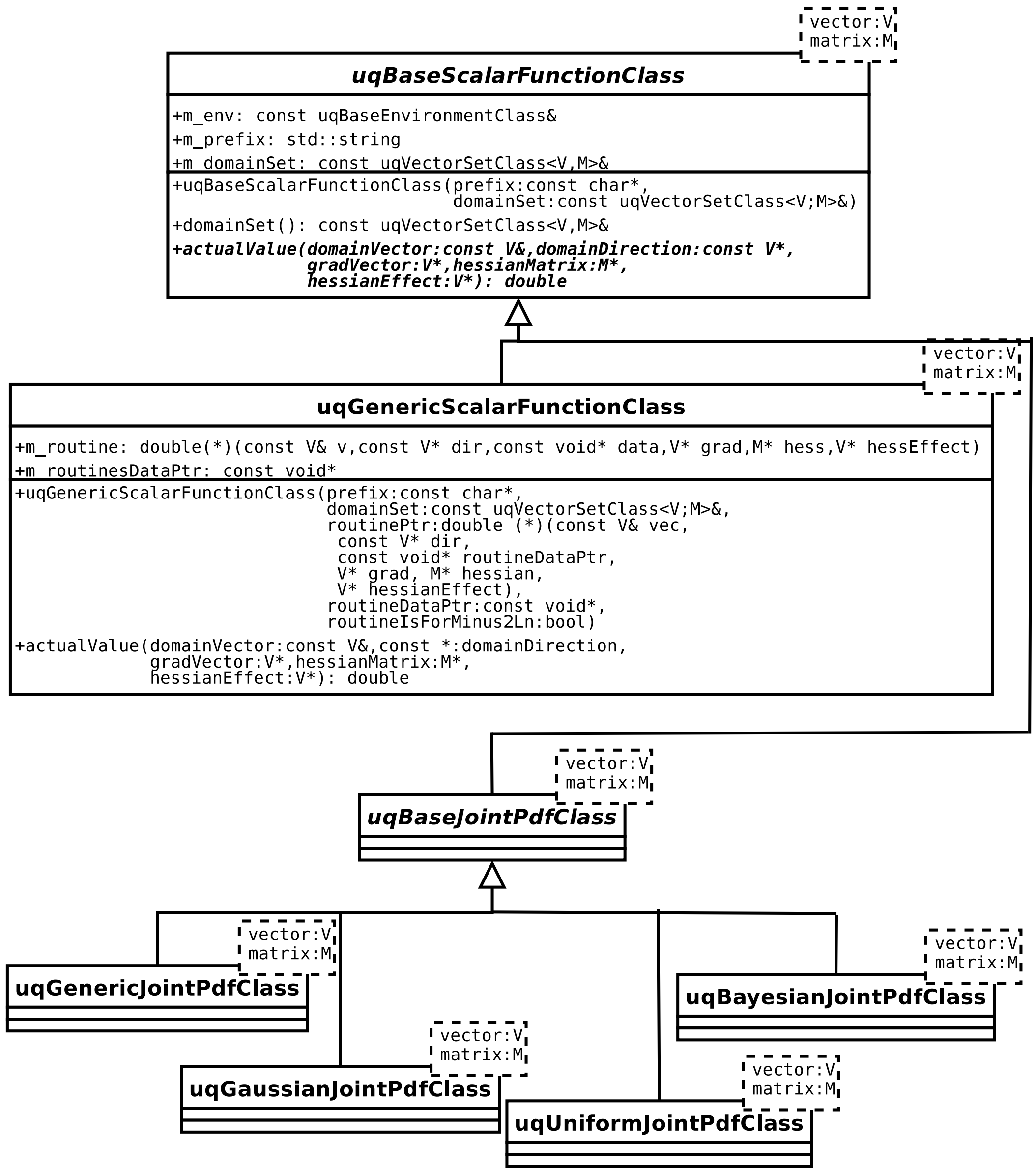

Figure 3.3.2: The class diagram for the scalar function class.

3.3.3 Vector Function

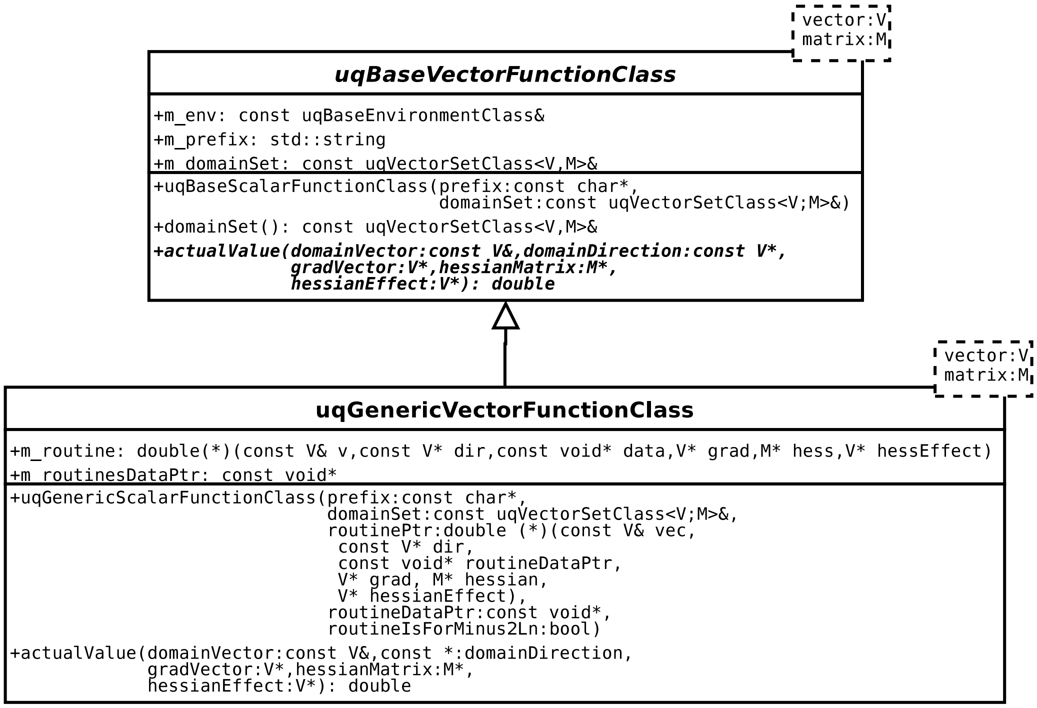

Figure 3.3.3: The class diagram for the vector function class.

3.3.4 Scalar Sequence

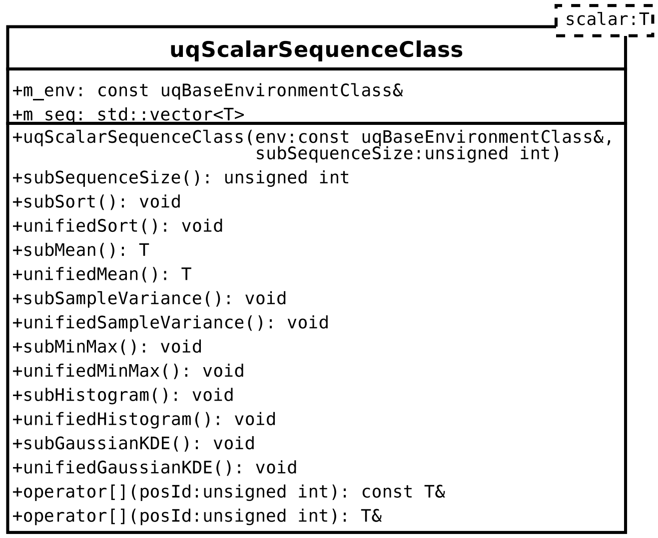

Figure 3.3.4: The class diagram for the scalar sequence class.

3.3.5 Vector Sequence

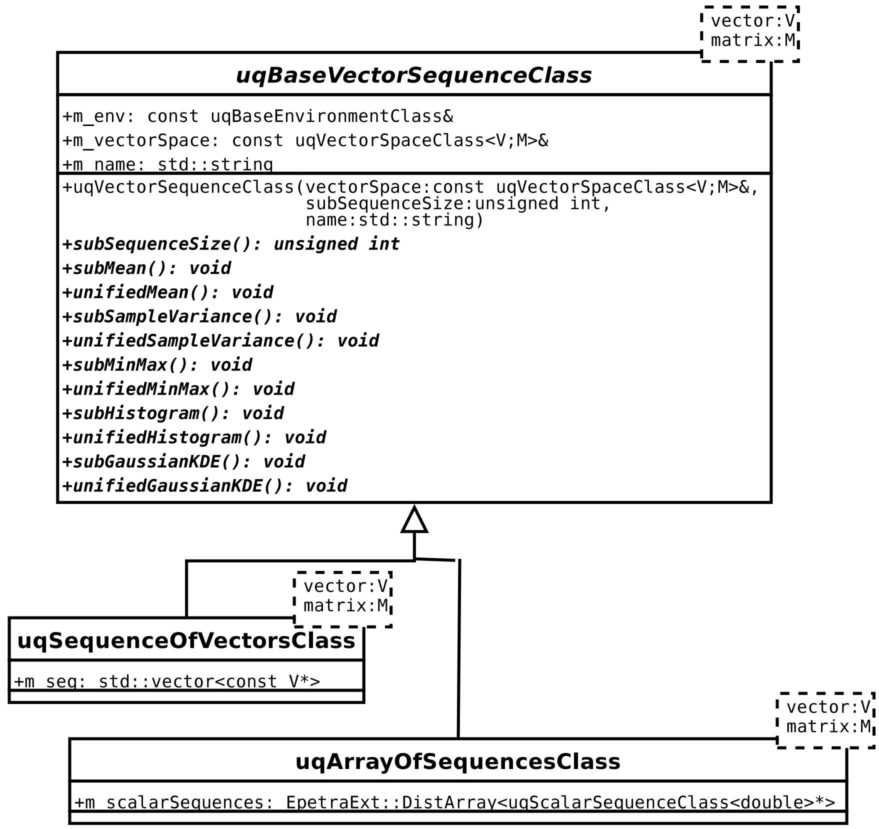

Figure 3.3.5: The class diagram for the vector sequence class.

3.4 Templated Statistical Classes

- Vector random variable
- Statistical inverse problem (and options)
- Metropolis-Hastings solver (and options)
- Statistical forward problem (and options)
- Monte Carlo solver (and options)
- Sequence statistical options

For QUESO, a statistical inverse problem has two input entities, a prior random variable and a likelihood routine, and one output entity, the posterior random variable, as shown in Figure 1.1.2.

Similarly, a statistical forward problem for QUESO has two input entities, a input random variable and a qoi routine, and one output entity, the output random variable, as shown in Figure 1.1.1.

3.4.1 Vector Random Variable

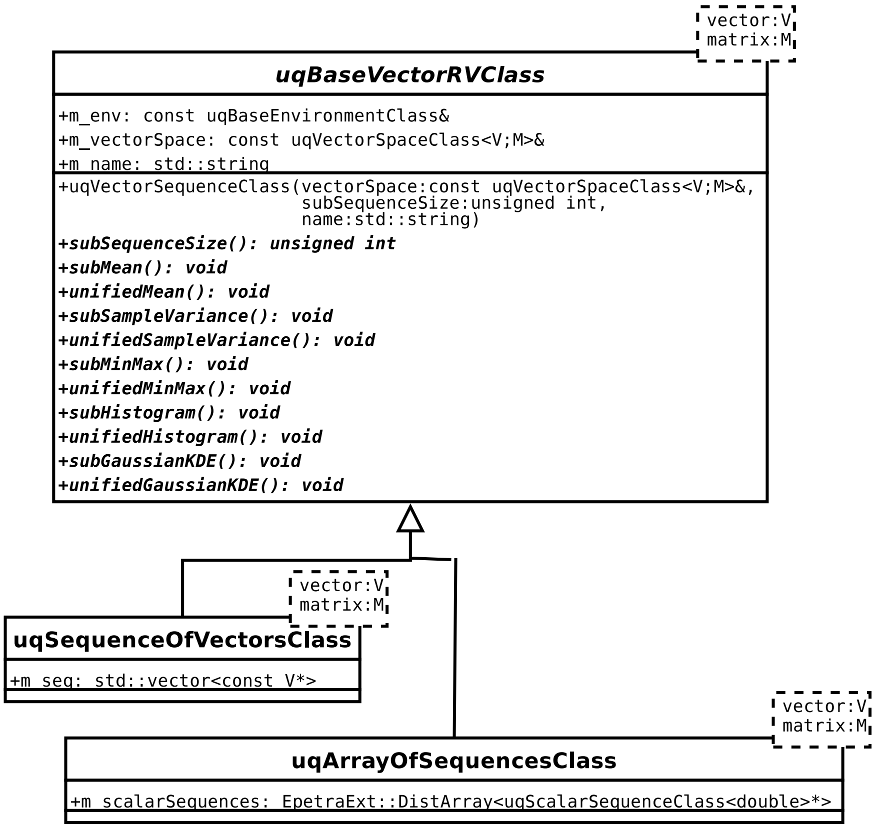

Figure 3.4.1: The class diagram for the vector random variable class.

3.4.2 Statistical Inverse Problem (and Options)

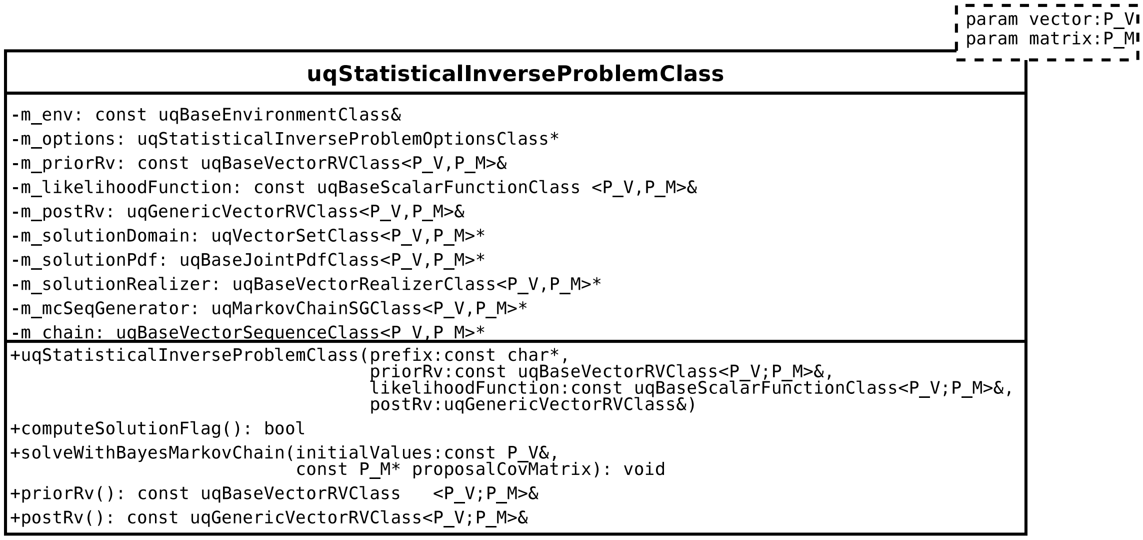

Figure 3.4.2: The statistical inverse problem class. It implements the representation in Figure 1.1.2.

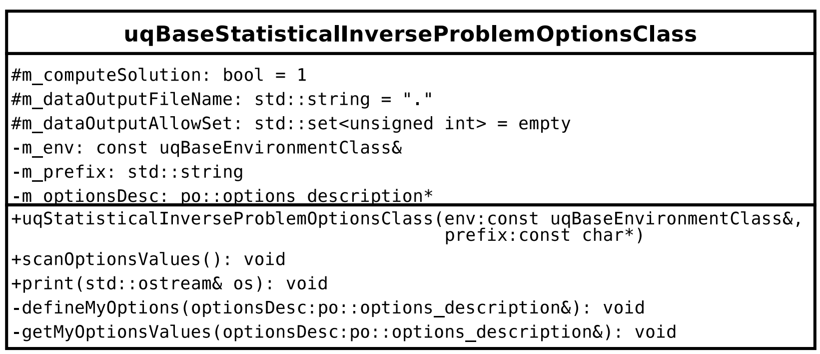

Figure 3.4.3: The statistical inverse problem options class.

Option Name	Default Value	Description
⟨PREFIX⟩ip_help		
⟨PREFIX⟩ip_computeSolution		
⟨PREFIX⟩ip_dataOutputFileName		
⟨PREFIX⟩ip_dataOutputAllowedSet		

Table 3.4.1: Input file options for a QUESO statistical inverse problem.

3.4.3 Metropolis-Hastings Solver (and Options)

```
                                                          param vector:P_V
                                                          param matrix:P_M
uqMetropolisHastingsSGClass
---------------------------------------------------------------------------
+m_env: const uqBaseEnvironmentClass&
+m_vectorSpace: const uqVectorSpaceClass<P_V,P_M>&
+m_targetPdf: const uqBaseJointPdfClass<P_V,P_M>&
+m_initialPosition: P_V
+m_initialProposalCovMatrix: const P_M*
+m_options: uqMetropolisHastingsSGOptionsClass*
---------------------------------------------------------------------------
+uqMetropolisHastingsSGClass(prefix:const char*,
                             sourceRv:const uqBaseVectorRVClass<P_V;P_M>&,
                             initialPosition:const P_V&,
                             inputProposalCovMatrix:const P_M*)
+generateSequence(workingChain:uqBaseVectorSequenceClass<P_V;P_M>&)
-generateFullChain(initialPosition:uqBaseVectorSequenceClass<P_V,
                   P_M>&)
+acceptAlpha(alpha:double): bool
```

Figure 3.4.4: The Metropolis-Hastings sequence generator class.

```
uqMetropolisHastingsSGOptionsClass
---------------------------------------------------------------------------
+m_dataOutputFileName: std::string = "."
+m_dataOutputAllowedSet: std::set<unsigned int> = empty
+m_rawChainSize: unsigned int = 100
+m_rawChainDisplayPeriod: unsigned int = 500
+m_rawChainDataOutputFileName: std::string
+m_rawChainDataOutputAllowedSet: std::set<unsigned int> = empty
+m_rawChainComputeStats: bool = false
+m_rawChainStatisticalOptions: uqChainStatisticalOptionsClass* = NULL
+m_filteredChainGenerate: bool = false
+m_filteredChainDiscardedPortion: double = 0
+m_filteredChainLag: unsigned int = 2
+m_filteredChainDataOutputFileName: std::string = "."
+m_filteredChainDataOutputAllowedSet: std::set<unsigned int> = empty
+m_filteredChainComputeStats: bool = false
+m_filteredChainStatisticalOptions: m_chainStatisticalOptionsClass* = NULL
+m_drMaxNumExtraStages: unsigned int
+m_drScalesForCovMatrices: std::vector<double>
+m_amInitialNonAdaptInterval: unsigned int
+m_amAdaptInterval: unsigned int
+m_amEta: double
+m_amEpsilon: double
-m_env: const uqBaseEnvironmentClass&
-m_prefix: std::string
-m_optionsDesc: po::options_description*
---------------------------------------------------------------------------
+uqMetropolisHastingsSGOptionsClass(env:const uqBaseEnvironmentClass&,
                                    prefix:const char*)
+scanOptionsValues(): void
+print(std::ofstream& os): void
-defineMyOptions(optionsDesc:po::options_description&): void
-getMyOptionsValues(optionsDesc:po::options_description&): void
```

Figure 3.4.5: The Metropolis-Hastings sequence generator options class.

Option Name	Default Value	Description
⟨PREFIX⟩mh_help		
⟨PREFIX⟩mh_dataOutputFileName		
⟨PREFIX⟩mh_dataOutputAllowedSet		
⟨PREFIX⟩mh_rawChain_dataInputFileName		
⟨PREFIX⟩mh_rawChain_size		
⟨PREFIX⟩mh_rawChain_generateExtra		
⟨PREFIX⟩mh_rawChain_displayPeriod		
⟨PREFIX⟩mh_rawChain_measureRunTimes		
⟨PREFIX⟩mh_rawChain_dataOutputFileName		
⟨PREFIX⟩mh_rawChain_dataOutputAllowedSet		
⟨PREFIX⟩mh_rawChain_computeStats		
⟨PREFIX⟩mh_filteredChain_generate		
⟨PREFIX⟩mh_filteredChain_discardedPortion		
⟨PREFIX⟩mh_filteredChain_lag		
⟨PREFIX⟩mh_filteredChain_dataOutputFileName		
⟨PREFIX⟩mh_filteredChain_dataOutputAllowedSet		
⟨PREFIX⟩mh_filteredChain_computeStats		
⟨PREFIX⟩mh_displayCandidates		
⟨PREFIX⟩mh_putOutOfBoundsInChain		
⟨PREFIX⟩mh_tk_useLocalHessian		
⟨PREFIX⟩mh_tk_useNewtonComponent		
⟨PREFIX⟩mh_dr_maxNumExtraStages		
⟨PREFIX⟩mh_dr_scalesForExtraStages		
⟨PREFIX⟩mh_am_initialNonAdaptInterval		
⟨PREFIX⟩mh_am_adaptInterval		
⟨PREFIX⟩mh_am_eta		
⟨PREFIX⟩mh_am_epsilon		

Table 3.4.2: Input file options for a QUESO Metropolis-Hastings solver.

3.4.4 Statistical Forward Problem (and Options)

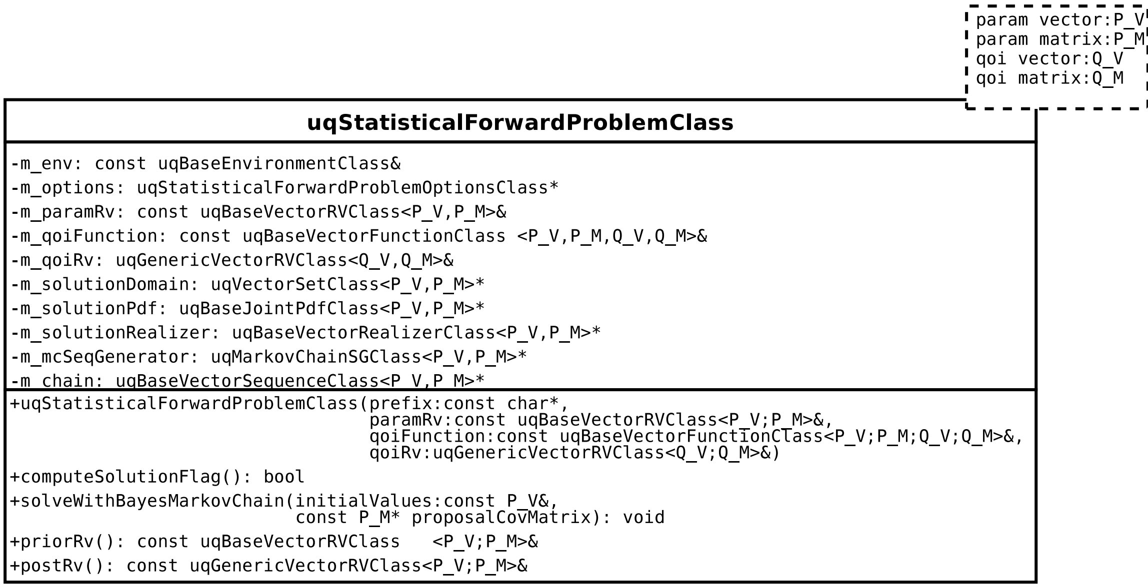

Figure 3.4.6: The statistical forward problem class. It implements the representation in Figure 1.1.1.

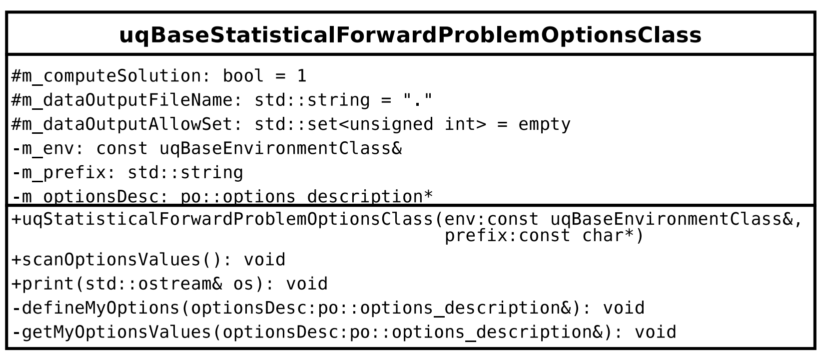

Figure 3.4.7: The statistical forward problem options class.

Option Name	Default Value	Description
⟨PREFIX⟩fp_help		
⟨PREFIX⟩fp_computeSolution		
⟨PREFIX⟩fp_computeCovariances		
⟨PREFIX⟩fp_computeCorrelations		
⟨PREFIX⟩fp_dataOutputFileName		
⟨PREFIX⟩fp_dataOutputAllowedSet		

Table 3.4.3: Input file options for a QUESO statistical forward problem.

3.4.5 Monte Carlo Solver (and Options)

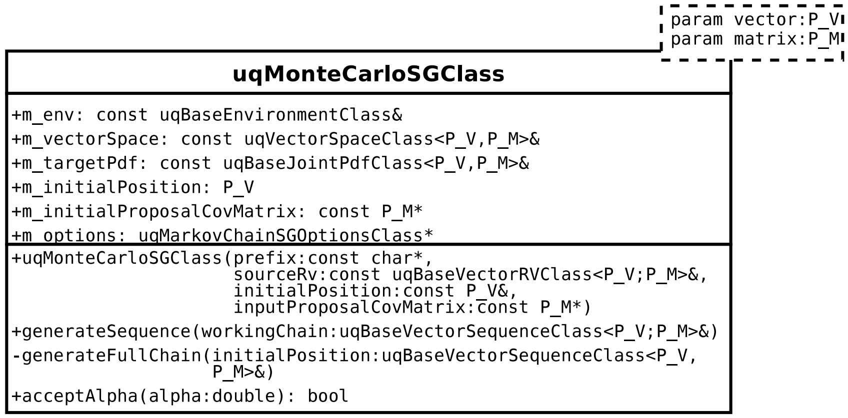

Figure 3.4.8: The Monte Carlo sequence generator class.

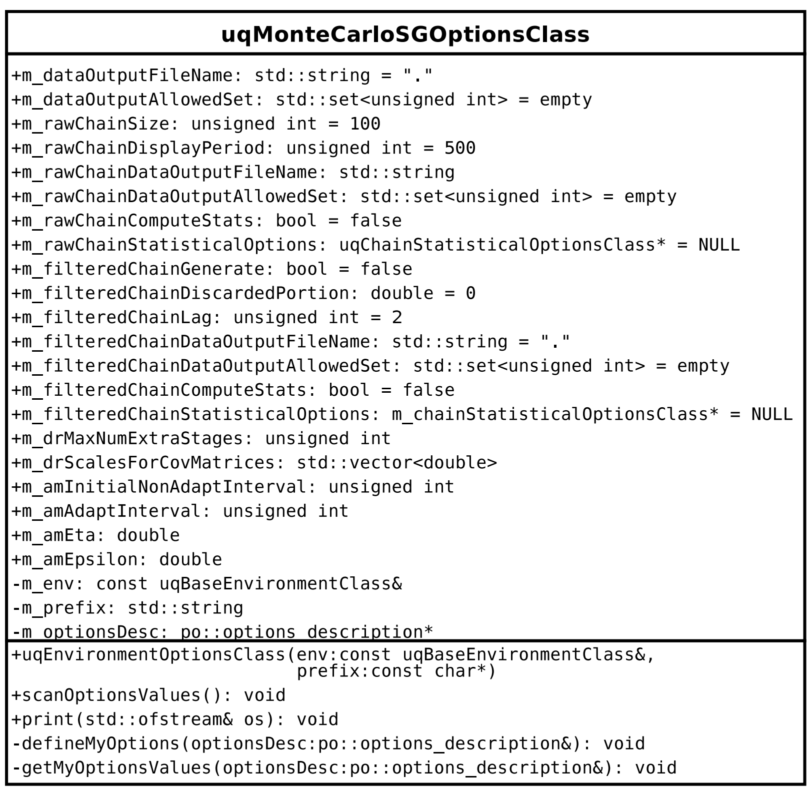

Figure 3.4.9: The Monte Carlo sequence generator options class.

Option Name	Default Value	Description
⟨PREFIX⟩mc_help		
⟨PREFIX⟩mc_dataOutputFileName		
⟨PREFIX⟩mc_dataOutputAllowedSet		
⟨PREFIX⟩mc_pseq_dataOutputFileName		
⟨PREFIX⟩mc_pseq_dataOutputAllowedSet		
⟨PREFIX⟩mc_pseq_computeStats		
⟨PREFIX⟩mc_qseq_dataInputFileName		
⟨PREFIX⟩mc_qseq_size		
⟨PREFIX⟩mc_qseq_displayPeriod		
⟨PREFIX⟩mc_qseq_measureRunTimes		
⟨PREFIX⟩mc_qseq_dataOutputFileName		
⟨PREFIX⟩mc_qseq_dataOutputAllowedSet		
⟨PREFIX⟩mc_qseq_computeStats		

Table 3.4.4: Input file options for a QUESO Monte Carlo solver.

3.4.6 Options for Statistical Analysis of Sequences

```
uqSequenceStatisticalOptionsClass
-----------------------------------------------------------------
+m_numSubEnvironments: unsigned int = 1
+m_subDisplayFileName: std::string = "."
+m_subDisplayAllowAll: bool = false
+m_subDisplayAllowSet: std::set<unsigned int>
+m_displayVerbosity: unsigned int = 2
+m_syncVerbosity: unsigned int = 0
+m_seed: int = 0
-m_env: const uqBaseEnvironmentClass&
-m_prefix: std::string
-m_optionsDesc: po::options_description*
-----------------------------------------------------------------
+uqSequenceStatisticalOptionsClass(env:const uqBaseEnvironmentClass&,
                                   prefix:const char*)
+scanOptionsValues(): void
+print(std::ofstream& os): void
-defineMyOptions(optionsDesc:po::options_description&): void
-getMyOptionsValues(optionsDesc:po::options_description&): void
```

Figure 3.4.10: The sequence statistical options class.

Option Name	Default Value	Description
⟨PREFIX⟩stats_help		
⟨PREFIX⟩stats_initialDiscardedPortions		
⟨PREFIX⟩stats_bmm_run		
⟨PREFIX⟩stats_bmm_lengths		
⟨PREFIX⟩stats_bmm_display		
⟨PREFIX⟩stats_bmm_write		
⟨PREFIX⟩stats_fft_compute		
⟨PREFIX⟩stats_fft_paramId		
⟨PREFIX⟩stats_fft_size		
⟨PREFIX⟩stats_fft_testInversion		
⟨PREFIX⟩stats_fft_write		
⟨PREFIX⟩stats_psd_compute		
⟨PREFIX⟩stats_psd_numBlocks		
⟨PREFIX⟩stats_psd_hopSizeRatio		
⟨PREFIX⟩stats_psd_paramId		
⟨PREFIX⟩stats_psd_write		
⟨PREFIX⟩stats_psdAtZero_compute		
⟨PREFIX⟩stats_psdAtZero_numBlocks		
⟨PREFIX⟩stats_psdAtZero_hopSizeRatio		
⟨PREFIX⟩stats_psdAtZero_display		
⟨PREFIX⟩stats_psdAtZero_write		
⟨PREFIX⟩stats_geweke_compute		
⟨PREFIX⟩stats_geweke_naRatio		
⟨PREFIX⟩stats_geweke_nbRatio		
⟨PREFIX⟩stats_geweke_display		
⟨PREFIX⟩stats_geweke_write		
⟨PREFIX⟩stats_autoCorr_computeViaDef		
⟨PREFIX⟩stats_autoCorr_computeViaFft		
⟨PREFIX⟩stats_autoCorr_secondLag		
⟨PREFIX⟩stats_autoCorr_lagSpacing		
⟨PREFIX⟩stats_autoCorr_numLags		
⟨PREFIX⟩stats_autoCorr_display		
⟨PREFIX⟩stats_autoCorr_write		
⟨PREFIX⟩stats_meanStacc_compute		
⟨PREFIX⟩stats_hist_compute		
⟨PREFIX⟩stats_hist_numInternalBins		
⟨PREFIX⟩stats_cdfStacc_compute		
⟨PREFIX⟩stats_cdfStacc_numEvalPositions		
⟨PREFIX⟩stats_kde_compute		
⟨PREFIX⟩stats_kde_numEvalPositions		
⟨PREFIX⟩stats_covMatrix_compute		
⟨PREFIX⟩stats_corrMatrix_compute		

Table 3.4.5: Input file options for a the statistical analysis of sequences

3.5 Interface Classes

Chapter 4

An Application Example (Out of Date)

In this chapter we show how to use QUESO in order to develop an application that solves an example statistical inverse problem and an example statistical forward problem, where the solution of the former serves as input to the later. Section 4.1 gives the mathematical formulation of both example problems. Section 4.2 shows the codes that translate the mathematical language into C++ using the QUESO classes and algorithms. Section 4.3 shows how to compile the code. Section 4.4 shows an example input file for QUESO classes and algorithms. Section 4.5 shows how to run the code. Finally, Section 4.6 shows how to plot figures using output data generated by the application.

4.1 Examples of Statistical Problems

4.1.1 Statistical Inverse Problem

In this example we have

$$\pi_{\text{prior}}(\boldsymbol{\theta}) \propto 1$$

and

$$\pi_{\text{like}}(\boldsymbol{\theta}) \propto e^{-\frac{1}{2}\left\{(\boldsymbol{\theta}-\boldsymbol{\mu})^T[\mathbf{C}^{-1}](\boldsymbol{\theta}-\boldsymbol{\mu})\right\}},$$

where

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \in \mathbb{R}^2,$$

$$\boldsymbol{\mu} = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \tag{4.1.1}$$

and

$$\mathbf{C} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.1.2}$$

The posterior pdf is then given by

$$\pi_{\text{post}}(\boldsymbol{\theta}) \propto e^{-\frac{1}{2}\left\{(\boldsymbol{\theta}-\boldsymbol{\mu})^T[\mathbf{C}^{-1}](\boldsymbol{\theta}-\boldsymbol{\mu})\right\}}. \tag{4.1.3}$$

It is clear that for such problem it is possible to analytically compute the exact posterior pdf as

$$\begin{aligned} \pi_{\text{post}}(\boldsymbol{\theta}) &= \frac{1}{4\pi} e^{-\frac{1}{2}\left\{(\boldsymbol{\theta}-\boldsymbol{\mu})^T[\mathbf{C}^{-1}](\boldsymbol{\theta}-\boldsymbol{\mu})\right\}} \\ &= \frac{1}{4\pi} e^{-\frac{1}{8}(\theta_1+1)^2-\frac{1}{2}(\theta_2-2)^2}, \end{aligned}$$

and to sample it through the formula

$$\boldsymbol{\mu} + \mathbf{C}^{1/2}\mathcal{N}(0, I),$$

where $\mathcal{N}(0, I)$ designates a Gaussian joint pdf of zero mean and unit covariance matrix and

$$\mathbf{C}^{1/2} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

Nonetheless, we will use QUESO to sample the posterior (4.1.3) with a Markov chain algorithm available in QUESO, and then check the marginal results for θ_1 and θ_2 against the analytical formulas

$$\begin{aligned} \pi_{\text{post}}(\theta_1) &= \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{8}(\theta_1+1)^2}, \\ \pi_{\text{post}}(\theta_1) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\theta_2-2)^2}. \end{aligned}$$

4.1.2 Statistical Forward Problem

Once the solution $\mathbf{\Theta}_{\rm post}$ of the statistical inverse problem is obtained, it is used for a statistical forward problem with a qoi function

$$\mathbf{q} : \mathbb{R}^2 \to \mathbb{R}.$$

In this example the Also, we have the very simple situation

$$\mathbf{q}(\boldsymbol{\theta}) = \theta_1 + \theta_2, \quad \forall \boldsymbol{\theta} \in \mathbb{R}^2. \tag{4.1.4}$$

Since the solution $\mathbf{Q}$ of this statistical forward problem is the sum of two rvs $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$, and since these two rvs are independent Gaussian rvs by assumption, we should have

$$E[\mathbf{Q}] = E[\mathbf{\Theta}_1] + E[\mathbf{\Theta}_2] = -1 + 2 = 1 \tag{4.1.5}$$

and

$$V[\mathbf{Q}] = V[\mathbf{\Theta}_1] + V[\mathbf{\Theta}_2] = 4 + 1 = 5 \tag{4.1.6}$$

where “E” and “V” indicate expectation and variance respectively.

4.2 Application Code

The program example given in this paper is compatible with version 0.41.0 of QUESO [?]. The source code for the example is composed of 7 files:

- example_main.C (Figure 4.2.1),
- example_likelihood.h and example_likelihood.C (Figures 4.2.2 and 4.2.3),
- example_qoi.h and example_qoi.C (Figures 4.2.4 and 4.2.5), and
- example_compute.h and example_compute.C (Figures 4.2.6, 4.2.7 and 4.2.8).

```
#include <example_compute.h>

int main(int argc, char* argv[])
{
  // Initialize environment
  MPI_Init(&argc,&argv);

  UQ_FATAL_TEST_MACRO(argc != 2,
                      UQ_UNAVAILABLE_RANK,
                      "main()",
                      "input file must be specified in command line
                       as argv[1], just after executable argv[0]");
  uqFullEnvironmentClass* env =
    new uqFullEnvironmentClass(MPI_COMM_WORLD,argv[1],"");

  // Compute
  compute(*env);

  // Finalize environment
  delete env;
  MPI_Finalize();

  return 0;
}
```

Figure 4.2.1: The example_main.C file.

```
#ifndef __EX_LIKELIHOOD_H__
#define __EX_LIKELIHOOD_H__

#include <uqGslMatrix.h>

struct
likelihoodRoutine_DataType
{
  const uqGslVectorClass* meanVector;
  const uqGslMatrixClass* covMatrix;
};

double likelihoodRoutine(
  const uqGslVectorClass& paramValues,
  const uqGslVectorClass* paramDirection,
  const void*             functionDataPtr,
  uqGslVectorClass*       gradVector,
  uqGslMatrixClass*       hessianMatrix,
  uqGslVectorClass*       hessianEffect);

#endif
```

Figure 4.2.2: The example_likelihood.h file

```
#include <example_likelihood.h>

double likelihoodRoutine(
  const uqGslVectorClass& paramValues,
  const uqGslVectorClass* paramDirection,
  const void*             functionDataPtr,
  uqGslVectorClass*       gradVector,
  uqGslMatrixClass*       hessianMatrix,
  uqGslVectorClass*       hessianEffect)
{
  // Just checking: the user, at the application level, expects
  // vector 'paramValues' to have size 2.
  UQ_FATAL_TEST_MACRO(paramValues.sizeGlobal() != 2,
                      UQ_UNAVAILABLE_RANK,
                      "likelihoodRoutine()",
                      "paramValues vector does not have size 2");

  // This code exemplifies multiple Metropolis-Hastings solvers, each calling
  // this likelihood routine.
  //
  // In this simple example, only node 0 in each subenvironment does the job
  // even though there might be more than one node per subenvironment.
  // In a more realistic situation, if the user is asking for multiple nodes per
  // subenvironment, then the model code in the qoi and likelihood routines
  // might really demand more than one node.
  //
  // Here we use 'env.subRank()' only. A realistic application might want to use
  // 'env.subComm()' or 'env.subComm().Comm()'
  double result = 0.;
  const uqBaseEnvironmentClass& env = paramValues.env();
  if (env.subRank() == 0) {
    const uqGslVectorClass& meanVector =
      *((likelihoodRoutine_DataType *) functionDataPtr)->meanVector;
    const uqGslMatrixClass& covMatrix  =
      *((likelihoodRoutine_DataType *) functionDataPtr)->covMatrix;

    uqGslVectorClass diffVec(paramValues - meanVector);

    result = scalarProduct(diffVec, covMatrix.invertMultiply(diffVec));
  }
  else {
    // Do nothing;
  }

  return -.5*result;
}
```

Figure 4.2.3: The example_likelihood.C file

```
#ifndef __EX_QOI_H__
#define __EX_QOI_H__

#include <uqGslMatrix.h>
#include <EpetraExt_DistArray.h>

struct
qoiRoutine_DataType
{
  double coef1;
  double coef2;
};

void
qoiRoutine(
  const uqGslVectorClass&                        paramValues,
  const uqGslVectorClass*                        paramDirection,
  const void*                                    functionDataPtr,
        uqGslVectorClass&                        qoiValues,
        EpetraExt::DistArray<uqGslVectorClass*>* gradVectors,
        EpetraExt::DistArray<uqGslMatrixClass*>* hessianMatrices,
        EpetraExt::DistArray<uqGslVectorClass*>* hessianEffects);

#endif
```

Figure 4.2.4: The example_qoi.h file

```
#include <example_qoi.h>
void qoiRoutine(
  const uqGslVectorClass&                       paramValues,
  const uqGslVectorClass*                       paramDirection,
  const void*                                   functionDataPtr,
        uqGslVectorClass&                       qoiValues,
        EpetraExt::DistArray<uqGslVectorClass*>* gradVectors,
        EpetraExt::DistArray<uqGslMatrixClass*>* hessianMatrices,
        EpetraExt::DistArray<uqGslVectorClass*>* hessianEffects)
{
  // Just checking: the user, at the application level, expects
  // vector 'paramValues' to have size 2 and
  // vector 'qoiValues' to have size 1.
  UQ_FATAL_TEST_MACRO(paramValues.sizeGlobal() != 2,
                      UQ_UNAVAILABLE_RANK,
                      "qoiRoutine()",
                      "paramValues vector does not have size 2");
  UQ_FATAL_TEST_MACRO(qoiValues.sizeGlobal() != 1,
                      UQ_UNAVAILABLE_RANK,
                      "qoiRoutine()",
                      "qoiValues vector does not have size 1");

  // This code exemplifies multiple Monte Carlo solvers, each calling this
  // qoi routine.
  //
  // In this simple example, only node 0 in each subenvironment does the job
  // even though there might be more than one node per subenvironment.
  // In a more realistic situation, if the user is asking for multiple nodes per
  // subenvironment, then the model code in the qoi and likelihood routines
  // might really demand more than one node.
  //
  // Here we use 'env.subRank()' only. A realistic application might want to use
  // 'env.subComm()' or 'env.subComm().Comm()'
  const uqBaseEnvironmentClass& env = paramValues.env();
  if (env.subRank() == 0) {
    double coef1 = ((qoiRoutine_DataType *) functionDataPtr)->coef1;
    double coef2 = ((qoiRoutine_DataType *) functionDataPtr)->coef2;
    qoiValues[0] = (coef1*paramValues[0] + coef2*paramValues[1]);
  }
  else {
    qoiValues[0] = 0.;
  }
  return;
}
```

Figure 4.2.5: The example_qoi.C file

```
#ifndef __EX_COMPUTE_H__
#define __EX_COMPUTE_H__

#include <uqEnvironment.h>

void compute(const uqFullEnvironmentClass& env);

#endif
```

Figure 4.2.6: The example_compute.h file

```
#include <example_compute.h>
#include <example_likelihood.h>
#include <example_qoi.h>
#include <uqGslMatrix.h>
#include <uqStatisticalInverseProblem.h>
#include <uqStatisticalForwardProblem.h>

void compute(const uqFullEnvironmentClass& env) {
  // Step 1 of 9: Instantiate the parameter space
  uqVectorSpaceClass<uqGslVectorClass,uqGslMatrixClass>
    paramSpace(env, "param_", 2, NULL);

  // Step 2 of 9: Instantiate the parameter domain
  uqGslVectorClass paramMins(paramSpace.zeroVector());
  paramMins.cwSet(-INFINITY);
  uqGslVectorClass paramMaxs(paramSpace.zeroVector());
  paramMaxs.cwSet( INFINITY);
  uqBoxSubsetClass<uqGslVectorClass,uqGslMatrixClass>
    paramDomain("param_",paramSpace,paramMins,paramMaxs);

  // Step 3 of 9: Instantiate the likelihood function object
  uqGslVectorClass meanVector(paramSpace.zeroVector());
  meanVector[0] = -1;
  meanVector[1] =  2;
  uqGslMatrixClass covMatrix(paramSpace.zeroVector());
  covMatrix(0,0) = 4.; covMatrix(0,1) = 0.;
  covMatrix(1,0) = 0.; covMatrix(1,1) = 1.;
  likelihoodRoutine_DataType likelihoodRoutine_Data;
  likelihoodRoutine_Data.meanVector = &meanVector;
  likelihoodRoutine_Data.covMatrix  = &covMatrix;
  uqGenericScalarFunctionClass<uqGslVectorClass,uqGslMatrixClass>
    likelihoodFunctionObj("like_",
                          paramDomain,
                          likelihoodRoutine,
                          (void *) &likelihoodRoutine_Data,
                          true); // routine computes [ln(function)]

  // Step 4 of 9: Instantiate the inverse problem
  uqUniformVectorRVClass<uqGslVectorClass,uqGslMatrixClass>
    priorRv("prior_", paramDomain);
  uqGenericVectorRVClass<uqGslVectorClass,uqGslMatrixClass>
    postRv("post_", paramSpace);
  uqStatisticalInverseProblemClass<uqGslVectorClass,uqGslMatrixClass>
    ip("", priorRv, likelihoodFunctionObj, postRv);
```

Figure 4.2.7: Initial part of example_compute.C file: the first 4 of the 5 steps to deal with the statistical inverse problem.

```
  // Step 5 of 9: Solve the inverse problem
  uqGslVectorClass paramInitials(paramSpace.zeroVector());
  paramInitials[0] = 0.1;
  paramInitials[1] = -1.4;
  uqGslMatrixClass proposalCovMatrix(paramSpace.zeroVector());
  proposalCovMatrix(0,0) = 8.; proposalCovMatrix(0,1) = 4.;
  proposalCovMatrix(1,0) = 4.; proposalCovMatrix(1,1) = 16.;
  ip.solveWithBayesMetropolisHastings(paramInitials, &proposalCovMatrix);

  // Step 6 of 9: Instantiate the qoi space
  uqVectorSpaceClass<uqGslVectorClass,uqGslMatrixClass>
    qoiSpace(env, "qoi_", 1, NULL);

  // Step 7 of 9: Instantiate the qoi function object
  qoiRoutine_DataType qoiRoutine_Data;
  qoiRoutine_Data.coef1 = 1.;
  qoiRoutine_Data.coef2 = 1.;
  uqGenericVectorFunctionClass<uqGslVectorClass,uqGslMatrixClass,
                               uqGslVectorClass,uqGslMatrixClass>
    qoiFunctionObj("qoi_",
                   paramDomain,
                   qoiSpace,
                   qoiRoutine,
                   (void *) &qoiRoutine_Data);

  // Step 8 of 9: Instantiate the forward problem
  uqGenericVectorRVClass<uqGslVectorClass,uqGslMatrixClass>
    qoiRv("qoi_", qoiSpace);
  uqStatisticalForwardProblemClass<uqGslVectorClass,uqGslMatrixClass,
                                   uqGslVectorClass,uqGslMatrixClass>
    fp("", postRv, qoiFunctionObj, qoiRv);

  // Step 9 of 9: Solve the forward problem
  fp.solveWithMonteCarlo();

  return;
}
```

Figure 4.2.8: Final part of example_compute.C file: the final step of the 5 steps to deal with the statistical inverse problem and the 4 steps to deal with the statistical forward problem.

4.3 Application Compilation

The makefile is given in Figure 4.3.1.

```
QUESO_DIR = /basepath/Installations/queso_0_41_0_gnu/
TRILINOS_DIR = /basepath/Installations/Trilinos_8_0_7/
BOOST_DIR = /basepath/Installations/Boost_1_35_0/
HPCT_DIR = /org/centers/pecos/LIBRARIES/hpct/0.25.1/
include $(TRILINOS_DIR)/include/Makefile.export.epetra

INC_PATHS = \
        -I. \
        -I$(QUESO_DIR)/include \
        -I$(MPI_DIR)/include \
        -I$(BOOST_DIR)/include/boost-1_35 \
        -I$(GSL_DIR)/include \
        -I$(HPCT_DIR)/include \
        $(EPETRA_INCLUDES)

LIBS = \
        -L$(QUESO_DIR)/lib -lqueso \
        -L$(MPI_DIR)/lib \
        -L$(TRILINOS_DIR)/lib \
        -L$(BOOST_DIR)/lib -lboost_program_options \
        -L$(GSL_DIR)/lib -lgsl \
        -L$(HPCT_DIR)/lib -lhpct \
        $(EPETRA_LIBS)

CXX = mpicxx
CXXFLAGS += -O3 -Wall -c

default: all

.SUFFIXES: .o .C

all:    ex_gsl

clean:
        rm -f *~
        rm -f *.o
        rm -f example

ex_gsl: example_main.o example_likelihood.o example_qoi.o example_compute.o
        $(CXX) example_main.o example_likelihood.o example_qoi.o example_compute.o \
               -o example_gsl $(LIBS)

%.o: %.C
        $(CXX) $(INC_PATHS) $(CXXFLAGS) $<
```

Figure 4.3.1: Makefile for the program in Figures 4.2.1-4.2.8.

4.4 Application Input File

The options input file is given in Figures 4.4.1, 4.4.2 and 4.4.3.

```
###############################################
# UQ Environment
###############################################
#env_help                 = anything
env_numSubEnvironments   = 1
env_subDisplayFileName   = outputData/display
env_subDisplayAllowAll   = 0
env_subDisplayAllowedSet = 0
env_displayVerbosity     = 2
env_syncVerbosity        = 0
env_seed                 = 0

###############################################
# Statistical inverse problem (ip)
###############################################
#ip_help                  = anything
ip_computeSolution      = 1
ip_dataOutputFileName   = outputData/sipOutput
ip_dataOutputAllowedSet = 0

###############################################
# Statistical forward problem (fp)
###############################################
fp_help                 = anything
fp_computeSolution      = 1
fp_computeCovariances   = 1
fp_computeCorrelations  = 1
fp_dataOutputFileName   = outputData/sfpOutput
fp_dataOutputAllowedSet = 0 1
```

Figure 4.4.1: Some options in the input file for program in Figures 4.2.1-4.2.8.

```
###############################################
# 'ip_': information for Metropolis-Hastings algorithm
###############################################
ip_mh_dataOutputFileName   = outputData/sipOutput
ip_mh_dataOutputAllowedSet = 0 1

ip_mh_rawChain_size                 = 32768
ip_mh_rawChain_dataOutputFileName   = outputData/ip_raw_chain
ip_mh_rawChain_dataOutputAllowedSet = 0 1
ip_mh_rawChain_computeStats         = 1

ip_mh_dr_maxNumExtraStages          = 1
ip_mh_dr_listOfScalesForExtraStages = 5.
ip_mh_am_initialNonAdaptInterval    = 0
ip_mh_am_adaptInterval              = 100
ip_mh_am_eta                        = 1.92
ip_mh_am_epsilon                    = 1.e-5

ip_mh_filteredChain_generate             = 1
ip_mh_filteredChain_discardedPortion     = 0.
ip_mh_filteredChain_lag                  = 16
ip_mh_filteredChain_dataOutputFileName   = outputData/ip_filt_chain
ip_mh_filteredChain_dataOutputAllowedSet = 0 1
ip_mh_filteredChain_computeStats         = 1

ip_mh_rawChain_stats_autoCorr_computeViaFft    = 1
ip_mh_rawChain_stats_autoCorr_secondLag        = 2
ip_mh_rawChain_stats_autoCorr_lagSpacing       = 2
ip_mh_rawChain_stats_autoCorr_numLags          = 10
ip_mh_rawChain_stats_autoCorr_display          = 1
ip_mh_rawChain_stats_autoCorr_write            = 1

ip_mh_filteredChain_stats_autoCorr_computeViaFft    = 1
ip_mh_filteredChain_stats_autoCorr_secondLag        = 2
ip_mh_filteredChain_stats_autoCorr_lagSpacing       = 2
ip_mh_filteredChain_stats_autoCorr_numLags          = 10
ip_mh_filteredChain_stats_autoCorr_display          = 1
ip_mh_filteredChain_stats_autoCorr_write            = 1
ip_mh_filteredChain_stats_hist_compute              = 1
ip_mh_filteredChain_stats_hist_numInternalBins      = 250
ip_mh_filteredChain_stats_kde_compute               = 1
ip_mh_filteredChain_stats_kde_numEvalPositions      = 250
ip_mh_filteredChain_stats_covMatrix_compute         = 1
ip_mh_filteredChain_stats_corrMatrix_compute        = 1
```

Figure 4.4.2: Options for the Markov chain algorithm for solving the statistical inverse problem.

```
###############################################
# 'fp_': information for Monte Carlo algorithm
###############################################
fp_mc_help                 = anything
fp_mc_dataOutputFileName   = outputData/sfpOutput
fp_mc_dataOutputAllowedSet = 0 1

fp_mc_pseq_dataOutputFileName   = outputData/fp_p_seq
fp_mc_pseq_dataOutputAllowedSet = 0 1
fp_mc_pseq_computeStats         = 1

#fp_mc_pseq_stats_help                     = anything
fp_mc_pseq_stats_initialDiscardedPortions  = 0.
fp_mc_pseq_stats_hist_compute              = 1
fp_mc_pseq_stats_hist_numInternalBins      = 250
fp_mc_pseq_stats_kde_compute               = 1
fp_mc_pseq_stats_kde_numEvalPositions      = 250
fp_mc_pseq_stats_covMatrix_compute         = 1
fp_mc_pseq_stats_corrMatrix_compute        = 1

fp_mc_qseq_size                 = 1048576
fp_mc_qseq_displayPeriod        = 20000
fp_mc_qseq_measureRunTimes      = 1
fp_mc_qseq_dataOutputFileName   = outputData/fp_q_seq
fp_mc_qseq_dataOutputAllowedSet = 0 1
fp_mc_qseq_computeStats         = 1

#fp_mc_qseq_stats_help                     = anything
fp_mc_qseq_stats_initialDiscardedPortions  = 0.
fp_mc_qseq_stats_autoCorr_computeViaFft    = 1
fp_mc_qseq_stats_autoCorr_secondLag        = 2
fp_mc_qseq_stats_autoCorr_lagSpacing       = 1
fp_mc_qseq_stats_autoCorr_numLags          = 15
fp_mc_qseq_stats_autoCorr_display          = 1
fp_mc_qseq_stats_autoCorr_write            = 1
fp_mc_qseq_stats_hist_compute              = 1
fp_mc_qseq_stats_hist_numInternalBins      = 250
fp_mc_qseq_stats_kde_compute               = 1
fp_mc_qseq_stats_kde_numEvalPositions      = 250
fp_mc_qseq_stats_covMatrix_compute         = 1
fp_mc_qseq_stats_corrMatrix_compute        = 1
```

Figure 4.4.3: Options for the Monte Carlo algorithm for solving the statistical forward problem.

4.5 Application Run

Once the code is compiled, one just needs to run “example_gsl example.inp”.

4.6 Application Results and Some Plots

The files generated will be in subdirectory "outputData", as specified by the options input file. Figures 4.6.1 and 4.6.2 show the contents of "test/t02_sip_sfp/sip_sfp/example_plots.m".

```
cd outputData
sipOutput_sub0
sfpOutput_sub0

plot(ip_mh_rawChain_corrViaFftLags_sub0,ip_mh_rawChain_corrViaFftInitPos0_sub0(1,:),'-b','li
hold
plot(ip_mh_filtChain_corrViaFftLags_sub0,ip_mh_filtChain_corrViaFftInitPos0_sub0(1,:),'-r','
ylabel('Autocorrelation for \theta_1','fontsize',20);
xlabel('Lag','fontsize',20);
a = axis;
axis([a(1) a(2) -0.1 1]);
grid minor;
set(gca,'fontsize',20);
legend('raw chain',...
       'filtered chain',...
       'location','northeast');
print -dpng paper_plot1.png
waitforbuttonpress;
clf;

plot(ip_mh_rawChain_corrViaFftLags_sub0,ip_mh_rawChain_corrViaFftInitPos0_sub0(2,:),'-b','li
hold
plot(ip_mh_filtChain_corrViaFftLags_sub0,ip_mh_filtChain_corrViaFftInitPos0_sub0(2,:),'-r','
ylabel('Autocorrelation for \theta_2','fontsize',20);
xlabel('Lag','fontsize',20);
a = axis;
axis([a(1) a(2) -0.1 1]);
grid minor;
set(gca,'fontsize',20);
legend('raw chain',...
       'filtered chain',...
       'location','northeast');
print -dpng paper_plot2.png
waitforbuttonpress;
clf;
```

Figure 4.6.1: Matlab program file for plotting: part 1 of 2

```
plot(ip_mh_filtChain_unifGkdePosits_sub0(1,:),ip_mh_filtChain_unifGkdeValues_sub0(1,:)
hold
x = ip_mh_filtChain_unifGkdePosits_sub0(1,:);
plot(x,(exp(-(x+1).*(x+1)/8))/2/sqrt(2*pi),'--r','linewidth',2);
ylabel('Posterior marginal pdf','fontsize',20);
xlabel('\theta_1','fontsize',20);
grid minor;
set(gca,'fontsize',20);
legend('QUESO',...
       'Analytic',...
       'location','northwest');
print -dpng paper_plot3.png
waitforbuttonpress;
clf;

plot(ip_mh_filtChain_unifGkdePosits_sub0(2,:),ip_mh_filtChain_unifGkdeValues_sub0(2,:)
hold
x = ip_mh_filtChain_unifGkdePosits_sub0(2,:);
plot(x,(exp(-(x-2).*(x-2)/2))/sqrt(2*pi),'--r','linewidth',2);
ylabel('Posterior marginal pdf','fontsize',20);
xlabel('\theta_2','fontsize',20);
%title('Fig 4, Pdfs for \theta_2','fontsize',20);
grid minor;
set(gca,'fontsize',20);
legend('QUESO',...
       'Analytic',...
       'location','northwest');
print -dpng paper_plot4.png
waitforbuttonpress;
clf;

plot(fp_mc_QoiSeq_unifGkdePosits_sub0(1,:),fp_mc_QoiSeq_unifGkdeValues_sub0(1,:),'-b',
ylabel('Pdf','fontsize',20);
xlabel('QoI = \theta_1 + \theta_2','fontsize',20);
a = axis;
axis([-9 11 a(3) a(4)]);
grid minor;
set(gca,'fontsize',20);
print -dpng paper_plot5.png
```

Figure 4.6.2: Matlab program file for plotting: part 2 of 2

4.6.1 Results for the Statistical Inverse Problem

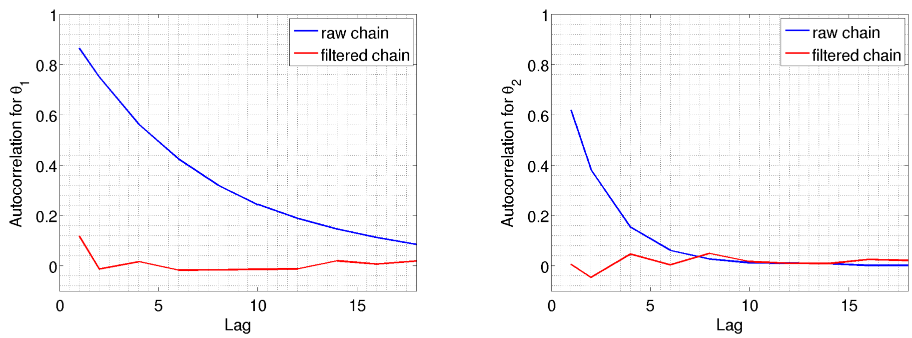

Figure 4.6.3: Autocorrelation plots obtained with QUESO for the statistical inverse problem. The user might want to see the results with a Markov chain with more positions than "just" 32,768. One might set "ip_mh_rawChain_size = 1,048,576" in the options input file, for instance.

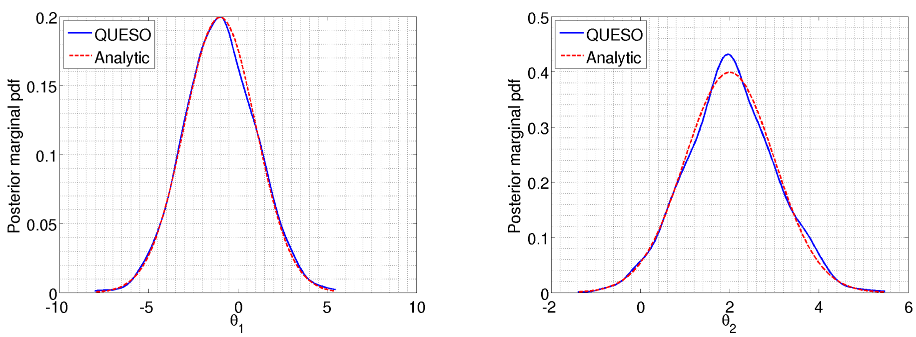

Figure 4.6.4: KDE plots obtained with QUESO for the statistical inverse problem. The user might want to see the results with a Markov chain with more positions than "just" 32,768. One might set "ip_mh_rawChain_size = 1,048,576" in the options input file, for instance.

4.6.2 Results for the Statistical Forward Problem

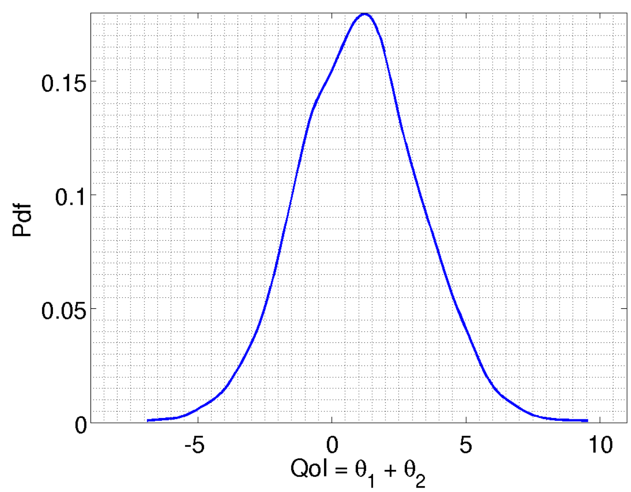

Figure 4.6.5: KDE plot obtained with QUESO for the statistical forward problem. The user might want to see the results with a Markov chain with more positions than "just" 32,768. One might set "ip_mh_rawChain_size = 1,048,576" in the options input file, for instance.

Appendix A

Free Software Needs Free Documentation

The following article was written by Richard Stallman, founder of the GNU Project.

The biggest deficiency in the free software community today is not in the software—it is the lack of good free documentation that we can include with the free software. Many of our most important programs do not come with free reference manuals and free introductory texts. Documentation is an essential part of any software package; when an important free software package does not come with a free manual and a free tutorial, that is a major gap. We have many such gaps today.

Consider Perl, for instance. The tutorial manuals that people normally use are non-free. How did this come about? Because the authors of those manuals published them with restrictive terms—no copying, no modification, source files not available—which exclude them from the free software world.

That wasn't the first time this sort of thing happened, and it was far from the last. Many times we have heard a GNU user eagerly describe a manual that he is writing, his intended contribution to the community, only to learn that he had ruined everything by signing a publication contract to make it non-free.

Free documentation, like free software, is a matter of freedom, not price. The problem with the non-free manual is not that publishers charge a price for printed copies—that in itself is fine. (The Free Software Foundation sells printed copies of manuals, too.) The problem is the restrictions on the use of the manual. Free manuals are available in source code form, and give you permission to copy and modify. Non-free manuals do not allow this.

The criteria of freedom for a free manual are roughly the same as for free software. Redistribution (including the normal kinds of commercial redistribution) must be permitted, so that the manual can accompany every copy of the program, both on-line and on paper.

Permission for modification of the technical content is crucial too. When people modify the software, adding or changing features, if they are conscientious they will change the manual too—so they can provide accurate and clear documentation for the modified program. A manual that leaves you no choice but to write a new manual to document a changed version of the program is not really available to our community.

Some kinds of limits on the way modification is handled are acceptable. For example, requirements to preserve the original author's copyright notice, the distribution terms, or the list of authors, are ok. It is also no problem to require modified versions to include notice that they were modified. Even entire sections that may not be deleted or changed are acceptable, as long as they deal with nontechnical topics (like this one). These kinds of restrictions are acceptable because they don't obstruct the community's normal use of the manual.

However, it must be possible to modify all the technical content of the manual, and then distribute the result in all the usual media, through all the usual channels. Otherwise, the restrictions obstruct the use of the manual, it is not free, and we need another manual to replace it.

Please spread the word about this issue. Our community continues to lose manuals to proprietary publishing. If we spread the word that free software needs free reference manuals and free tutorials, perhaps the next person who wants to contribute by writing documentation will realize, before it is too late, that only free manuals contribute to the free software community.

If you are writing documentation, please insist on publishing it under the GNU Free Documentation License or another free documentation license. Remember that this decision requires your approval—you don't have to let the publisher decide. Some commercial publishers will use a free license if you insist, but they will not propose the option; it is up to you to raise the issue and say firmly that this is what you want. If the publisher you are dealing with refuses, please try other publishers. If you're not sure whether a proposed license is free, write to lice

Appendix B

GNU General Public License

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions

0. Definitions.

   "This License" refers to version 3 of the GNU General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

   "The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

   To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

   A "covered work" means either the unmodified Program or a work based on the Program.

   To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

   The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

   The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

   The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

   All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

   You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

   Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

   You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

(a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

(b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

(c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

(d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

(a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

(b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

(c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

(d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

(e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

   "Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

   When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

   Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

   (a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
   (b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
   (c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

(d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

(e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

(f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

   You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

    You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

    A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

    A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

    Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

    If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

    Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

    If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

    Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

    THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

End of Terms and Conditions

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program’s name and a brief idea of what it does.>

Copyright (C) <textyear>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see `http://www.gnu.org/licenses/`.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read `http://www.gnu.org/philosophy/why-not-lgpl.html`.

Appendix C

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of

Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

[1] Boost Community. Boost C++ Libraries. http://www.boost.org/, 1998-2009.

[2] Mark Galassi et al. GNU Scientific Library. http://www.gnu.org/software/gsl/, 1996-2009.

[3] Michael Heroux. Trilinos. http://www.trilinos.gov/, 2009.

[4] Mpich. Message passing interface. http://www.mcs.anl.gov/mpi/, 1993-2009.

[5] Openmpi. Message passing interface. http://www.open-mpi.org/, 2004-2009.

[6] Karl Schulz. High Performance Computing Toolkit. User's Manual, 2008-2009.