MPI Implementation of some eigen libraries others

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 Genfun::Argument Class Reference

Public Member Functions

- **Argument** (int ndim=0)
- **Argument** (const Argument &)
- **Argument** (std::initializer_list< double >)
- const Argument & **operator=** (const Argument &)
- double & **operator[ ]** (int I)
- const double & **operator[ ]** (int i) const
- unsigned int **dimension** () const

Friends

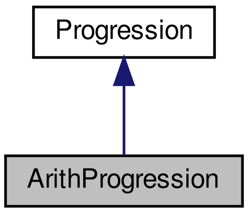
- std::ostream & **operator**<< (std::ostream &o, const Argument &a)

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

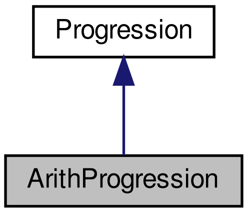
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Argument.h

3.2 ArithProgression Class Reference

Inheritance diagram for ArithProgression:



Collaboration diagram for ArithProgression:



Public Member Functions

- **ArithProgression** (long i=1)

Protected Member Functions

- virtual long **nextValue** ()

Protected Attributes

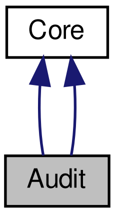
- long **inc**

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

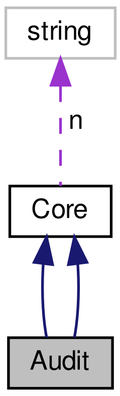
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

3.3 Audit Class Reference

Inheritance diagram for Audit:



Collaboration diagram for Audit:



Public Member Functions

- **Audit** (std::istream &is)
- std::istream & **read** (std::istream &)
- double **grade** () const
- bool **valid** () const
- bool **fulfill_reqs** () const
- **Audit** (std::istream &is)
- std::istream & **read** (std::istream &)
- double **grade** () const
- bool **valid** () const
- bool **fulfill_reqs** () const

Additional Inherited Members

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Core.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp_dynamicbindingandinheritance.hpp

3.4 background_task Class Reference

Public Member Functions

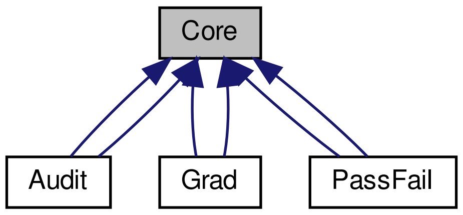
- void **operator()** () const

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

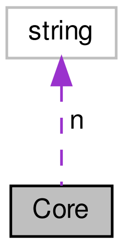
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp_thread.cxx

3.5 Core Class Reference

Inheritance diagram for Core:



Collaboration diagram for Core:



Public Member Functions

- **Core** (std::istream &is)
- std::string **name** () const
- virtual std::istream & **read** (std::istream &)
- virtual double **grade** () const
- virtual bool **valid** () const
- virtual bool **fulfill_reqs** () const
- **Core** (std::istream &is)
- std::string **name** () const
- virtual std::istream & **read** (std::istream &)
- virtual double **grade** () const
- virtual bool **valid** () const
- virtual bool **fulfill_reqs** () const

Protected Member Functions

- std::istream & **read_common** (std::istream &)
- virtual Core ∗ **clone** () const
- std::istream & **read_common** (std::istream &)
- virtual Core ∗ **clone** () const

Protected Attributes

- std::string **n**
- double **midterm**
- double **final**
- std::vector< double > **homework**

Friends

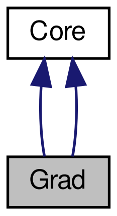
- class **Student_info**

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

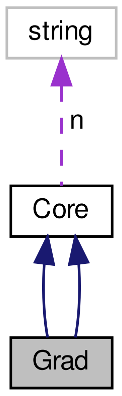
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Core.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp_dynamicbindingandinheritance.hpp

3.6 Grad Class Reference

Inheritance diagram for Grad:



Collaboration diagram for Grad:



Public Member Functions

- **Grad** (std::istream &is)
- std::istream & **read** (std::istream &)
- double **grade** () const
- bool **fulfill_reqs** () const
- **Grad** (std::istream &is)
- std::istream & **read** (std::istream &)
- double **grade** () const
- bool **fulfill_reqs** () const

Additional Inherited Members

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Core.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp_dynamicbindingandinheritance.hpp

3.7 InitiateVectorMethod$<$ ItemType $>$ Class Template Reference

Public Member Functions

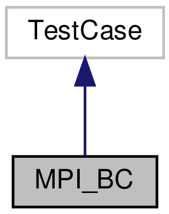
- **InitiateVectorMethod** (int, int)
- void **setup** (int $*$)
- void **traits** ()
- void **SendVector** ()
- void **GetData** ()

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

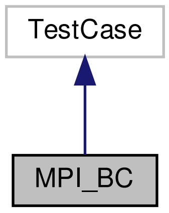
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/new.hpp

3.8 MPI_BC Class Reference

Inheritance diagram for MPI_BC:



Collaboration diagram for MPI_BC:



Public Member Functions

- MPI_BC ()

  MPI_BC class template filling.
- void **Get_input** (int, int, double ∗, double ∗, int ∗)
- void **packData** ()
- void **time_ellapsed** ()
- void **broadcast_input** ()
- void **broadcast_vector** ()
- void **buildMpiType** (double ∗, double ∗, int ∗, MPI_Datatype ∗)
- void **Send** (float, float, int, int)
- void **SendVector** ()
- void **Receive** (float ∗, float ∗, int ∗, int)
- void **parallelAllocateVec** (double ∗, double ∗, int, std::vector< int > ∗, MPI_Datatype ∗)

3.8.1 Constructor & Destructor Documentation

3.8.1.1 MPI_BC()

```
MPI_BC::MPI_BC ( )
```

MPI_BC class template filling.

Placeholder

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI_broadcast.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI_broadcast.cxx

3.9 MPI_BC_Generic$<$ T, Q, R $>$ Class Template Reference

Public Member Functions

- **MPI_BC_Generic** (std::size_t n)

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI_broadcast.hpp

3.10 MPI_sorting_methods Class Reference

Public Member Functions

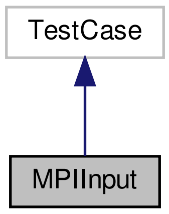
- void **Bubble_sort** (int a[ ], int n)

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

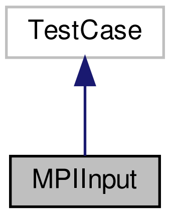
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI_reduce.cxx

3.11 MPIInput Class Reference

Inheritance diagram for MPIInput:



Collaboration diagram for MPIInput:



Public Member Functions

- MPIInput (int, int)

  MPIInput class - constructor.
- void **getData** (int ∗, int ∗, int ∗)
- void **bubbleSort** (int ∗, int)
- void getDataPack (float ∗, float ∗, int ∗)
- ∼MPIInput ()

  Class destructor for MPI.
- void **test1** ()

3.11.1 Constructor & Destructor Documentation

3.11.1.1 MPIInput()

```
MPIInput::MPIInput (
             int mr,
             int pe )
```

MPIInput class - constructor.

int int constructor

Sets up the processor ranks and size for use in other functions

At the moment, we do not have a default constructor

3.11.1.2 ∼MPIInput()

```
MPIInput::∼MPIInput ( )
```

Class destructor for MPI.

Destructor

MPI_Finalize() - What does it do?

3.11.2 Member Function Documentation

3.11.2.1 getDataPack()

```
void MPIInput::getDataPack (
             float * a_ptr,
             float * b_ptr,
             int * n_ptr )
```

< Placeholder

< Keeping track of where the data is

MPI_pack

MPI_pack allows one to explicitly store uncontiguous data in contiguous memory locations, and MPI_unpack can be used to copy data from a contiguous buffer into noncontiguous memory locations.

< Now pack the data into buffer. Positon = 0 says start at beginning of buffer

MPI_unpack

MPI_unpack can be used to copy data from a contiguous buffer into noncontiguous memory locations. i.e. it is just the complete opposite of pack

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI_IO.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI_IO.cxx

3.12 OMP< T > Class Template Reference

Openmp explanation.

```
#include <openmp1.hpp>
```

Public Member Functions

- **OMP** (int)
- **OMP** (const OMP &OMPCopy)
- OMP & **operator=** (const OMP &ref)
- void **add** (T)
- void **addup** ()
- void **pi** ()

3.12.1 Detailed Description

template<class T>
class OMP< T >

Openmp explanation.

As we noted earlier, we'll usually specify the number of threads on the command line, so we'll modify parallel directive was with the num_threads clause. A clause in OpenMP is just some text that modifies a directive. The num_threaads clause, it allows a programmer to specify the number of threads that should be executed in the following block:

pragma omp parallel num_thrads (thread_count)

What actually happens when the program gets to the parallel directive? Prior to the parallel directive, the program is using a single thrad, the process started when tbe program started execution.

< std::cout, std::endl

< std::size_t

< std::max

< std::allocator, std::uninitialized_fill, std::uninitialized_copy

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp1.hpp

3.13 Partstruct Struct Reference

Public Attributes

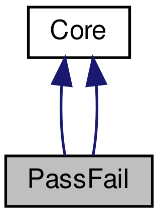
- int **class**
- double **d** [6]
- char **b** [7]

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

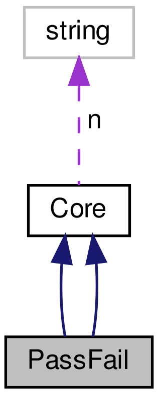
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI_struct.cxx

3.14 PassFail Class Reference

Inheritance diagram for PassFail:



Collaboration diagram for PassFail:



Public Member Functions

- **PassFail** (std::istream &is)
- double **grade** () const
- bool **valid** () const
- bool **fulfill_reqs** () const
- **PassFail** (std::istream &is)
- double **grade** () const
- bool **valid** () const
- bool **fulfill_reqs** () const

Additional Inherited Members

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Core.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp_dynamicbindingandinheritance.hpp

3.15 part1::Point Class Reference

```
#include <lib_mpi.hpp>
```

Public Member Functions

- **Point** (float _x, float _y, float _z)

Public Attributes

- float **x**
- float **y**
- float **z**

3.15.1 Detailed Description

This is a simple 3D point class

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/lib_mpi.hpp

3.16 ProbDist Class Reference

```
#include <openmp_LA.hpp>
```

Public Member Functions

- double **norm_pdf** (const double &x)
- double **norm_cdf** (const double &x)
- double **d_j** (const int &, const double &, const double &, const double &, const double &, const double &)
- double **call_price** (const double &, const double &, const double &, const double &, const double &)
- double **put_price** (const double &, const double &, const double &, const double &, const double &)
- double **gaussian_box_muller** ()
- double **monte_carlo_call_price** (const int &, const double &, const double &, const double &, const double &, const double &)

3.16.1 Detailed Description

European Options with Monte Carlo

In this chapter, we will pric a European Vanilla option via the correct analyci solution of the Black-Scholes eqiation, as well as via the Monte Carlo method. We won't be cooncentrating on an extremely efficient or optimised implementation at this stage.

-> Black Scholes Analytic pricing formula

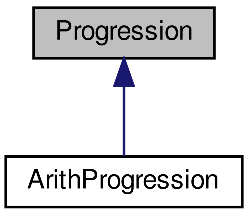
The first stage in implementation is to briefly discuess the Black Scholes analytic solution for the price of a vanilla call or put option

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp_LA.hpp

3.17 Progression Class Reference

Inheritance diagram for Progression:



Public Member Functions

- **Progression** (long f=0)
- void **printProgression** (int n)

Protected Member Functions

- virtual long **firstValue** ()
- virtual long **nextValue** ()

Protected Attributes

- long **first**
- long **cur**

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

3.18 QTstyle_Test Class Reference

A test class - find and replace from this template for future class definitions.

```
#include <statistics.h>
```

3.18.1 Detailed Description

A test class - find and replace from this template for future class definitions.

One of the most common examples of concepts in quantitiative finance is that of a statistical distribtion. Random variables play a huge part in quantitive financial modelling. Derivatives, pricing, cash-flow forceasting and quantitive trading all make use of statitiscal methods in some fashion

Many of the chapters within this book have made use of random number generators in order to carry out pricing tasks.

In a nutshell, we are splitting the generation of (uniform integer) random numbers from draws of specific statistical distribution,s such taht we can use the statics classes elsewhere withut bringing along the heavy random number generation functions.

Equally useful is the fact taht we will be able to "swap out" different random number generators for out statisics classes for reliability, extensibility and efficiency

A more elaborate class definition

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/statistics.h

3.19 RandomNumberGenerator Class Reference

Public Member Functions

- **RandomNumberGenerator** (unsigned long _num_draws, unsigned long _init_seed)
- virtual unsigned long **get_random_seed** () const
- virtual void **set_random_seed** (unsigned long _seed)
- virtual void **set_num_draws** (unsigned long _num_draws)
- virtual unsigned long **get_random_integer** ()=0

Protected Attributes

- unsigned long **init_seed**
- unsigned long **cur_seed**
- unsigned long **num_draws**

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/random.hpp

3.20 Stack< T, CONT > Class Template Reference

Public Member Functions

- void **push** (T const &)
- void **pop** ()
- T **top** () const
- bool **empty** () const

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

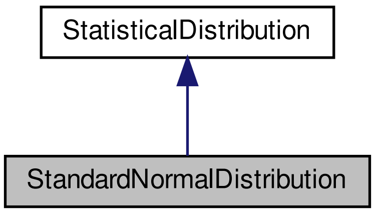
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp_templates.hpp

3.21 StandardNormalDistribution Class Reference

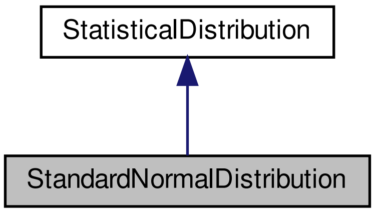
Standard Normal Distribution Implementation.

```
#include <statistics.h>
```

Inheritance diagram for StandardNormalDistribution:



Collaboration diagram for StandardNormalDistribution:



Public Member Functions

- virtual double **pdf** (const double &x) const
- virtual double **cdf** (const double &x) const
- virtual double **inv_cdf** (const double &quantile) const
- virtual double **mean** () const
- virtual double var () const

  Equal to 0.
- virtual double stddev () const

  Equal to 1.
- virtual void random_draws (const std::vector< double > &uniform_draws, std::vector< double > &dist_↩ draws)

  Variable 1.

3.21.1 Detailed Description

Standard Normal Distribution Implementation.

A more elaborate explanation here

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

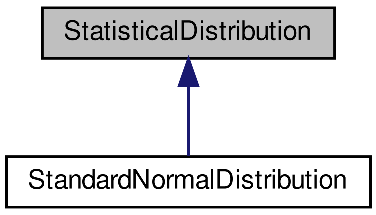
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/statistics.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/statistics.cxx

3.22 StatisticalDistribution Class Reference

Statistical Distribution Class.

```
#include <statistics.h>
```

Inheritance diagram for StatisticalDistribution:



Public Member Functions

- StatisticalDistribution ()

  A constructor.
- virtual ~StatisticalDistribution ()

  Virtual destructor.
- virtual double **pdf** (const double &x) const =0
- virtual double **cdf** (const double &x) const =0
- virtual double **inv_cdf** (const double &quantile) const =0
- virtual double **mean** () const =0
- virtual double var () const =0

  Variable 1.
- virtual double stdev () const =0

  Varable 2.
- virtual void random_draws (const std::vector< double > &uniform_draws, std::vector< double > &dist_↩ draws)=0

  Variable 3.

3.22.1 Detailed Description

Statistical Distribution Class.

We've specified pure virtual methods for the probability density function (pdf), cumulative density function (cdf), inverse cdf (inv_cdf), as well as descriptive statistics functions such as as mean, var (variance) and stdev.

Finally, we have a method that takes in a vector of uniform random variables on the open interval (0,1), then fills a vector of identical length with draws from the distribution

3.22.2 Constructor & Destructor Documentation

3.22.2.1 StatisticalDistribution()

```
StatisticalDistribution::StatisticalDistribution ( )
```

A constructor.

Statistical Distribution.

Statistical Distribution constructor

A more elaborate class definition

3.22.2.2 ∼StatisticalDistribution()

```
StatisticalDistribution::∼StatisticalDistribution ( )  [virtual]
```

Virtual destructor.

Constructor.

A more elaborate explanation here

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/statistics.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/statistics.cxx

3.23 Str Class Reference

A constructor with a character pointer input.

```
#include <openmp2.hpp>
```

Public Types

- typedef Vec< char >::size_type **size_type**

Public Member Functions

- Str ()
- **Str** (size_type n, char c)
- Str (const char ∗cp)

  Last two custom constructors.
- template<class In >
  Str (In b, In e)

  create a Str from the range denoted by iterators b and e

3.23.1 Detailed Description

A constructor with a character pointer input.

Custom Str class

A numeric progression is a seuqence of numbers, where the value of each number depends on one or more of the previous value.

Objects of built-in types generally behave like values: Whenever we copy an object of such a type, the original and copy have the same value but are otherwise indepedent.

For most of the built-in types, the language also defines a rich set of operators and provides automatic conversions between logically similar types. For example, if we add an int and a double, the compiler automatically converts the int into a double

When we define our own classes, we control the extent to which the resulting objects behave like values. By defining copying and assigning appropriately, the class author an arrange for objects of that class to act like values - that is, the class author can arrange for each object to have state that is independent of any other object.

Our Vec and Student_info classes are examples of types that act like values

We shall see that the class author an also control conversions and related operations on class objects, thereby providing classes whose objects behave even more similarly to objects of built-in types.

Defining a Str class that lets us create objects that behave approproximately as we would like.

3.23.2 Constructor & Destructor Documentation

3.23.2.1 Str() [1/3]

```
Str::Str ( )  [inline]
```

default constructor, create an empty str

3.23.2.2 Str() [2/3]

```
Str::Str (
             const char * cp )  [inline]
```

Last two custom constructors.

create a Str containing n copies of c

The last two consturctors are similar to each oter. Their constructor initializers are empty, which means that data is implicitly initializes as an empty Vec. Each constructor asks copy to append the supplied characters to the initally empty data.

3.23.2.3 Str() [3/3]

```
template<class In >
Str::Str (
            In b,
            In e )  [inline]
```

create a Str from the range denoted by iterators b and e

The most interesting constructor is the ifnal one, which takes two iterators and creates a new Str that contains a copy of the characters in the given sequence.

What is interesting about this constructor is that it is itself a template function. Because it is a template, it effectively defines a family of constrctors that can be instantiated for different types of iterators.

This constructor could be used to create a Str from an array of characters.

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

3.24 Student_info Class Reference

Public Member Functions

- **Student_info** (std::istream &is)
- **Student_info** (const Student_info &)
- Student_info & **operator=** (const Student_info &)
- std::istream & **read** (std::istream &)
- std::string **name** () const
- double **grade** () const

Static Public Member Functions

- static bool **compare** (const Student_info &s1, const Student_info &s2)

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Student_info.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/Student_info.cc
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/Student_info.cxx

3.25 TemplateUnderTest< T > Class Template Reference

Public Member Functions

- **TemplateUnderTest** (T ∗t)
- void **SomeMethod** ()

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

3.26 Trap Class Reference

Public Member Functions

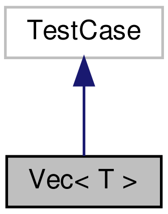
- void **read** ()
- void **computeTrapezium** ()

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

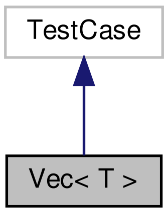
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/trapezoid.hpp

3.27 Vec$<$ T $>$ Class Template Reference

Inheritance diagram for Vec$<$ T $>$:



Collaboration diagram for Vec$<$ T $>$:



Public Types

- typedef T ∗ **iterator**
- typedef const T ∗ **const_iterator**
- typedef size_t **size_type**
- typedef T ∗ **iterator**
- typedef const T ∗ **const_iterator**
- typedef size_t **size_type**
- typedef T **value_type**
- typedef T & **reference**
- typedef const T & **const_reference**

Public Member Functions

- **Vec** (size_type n, const T &t=T())
- **Vec** (const Vec &v)
- Vec & **operator=** (const Vec &)
- const T & **operator[ ]** (size_type i) const
- void **push_back** (const T &t)
- size_type **size** () const
- iterator **begin** ()
- const_iterator **begin** () const
- iterator **end** ()
- const_iterator **end** () const
- void **runTest** ()

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI_str.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp1.hpp

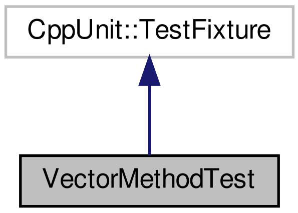
3.28 tutorial::Vec< T > Class Template Reference

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

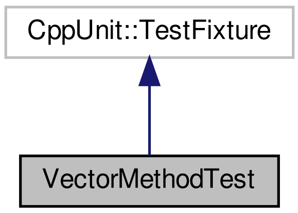
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/lib_mpi.hpp

3.29 VectorMethodTest Class Reference

Inheritance diagram for VectorMethodTest:



Collaboration diagram for VectorMethodTest:



Public Member Functions

- void **setUp** ()
- void **tearDown** ()
- void **testConstructor** ()

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/new.hpp

Index