

MPI Implementation of some eigen libraries others

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	Genfun::Argument Class Reference	5
3.2	ArithProgression Class Reference	5
3.3	Audit Class Reference	6
3.4	background_task Class Reference	7
3.5	Complex Class Reference	7
3.6	Core Class Reference	8
3.7	Grad Class Reference	9
3.8	InitiateVectorMethod< ItemType > Class Template Reference	10
3.9	MPI_BC Class Reference	10
3.9.1	Constructor & Destructor Documentation	11
3.9.1.1	MPI_BC()	11
3.10	MPI_BC_Generic< T, Q, R > Class Template Reference	11
3.11	MPI_sorting_methods Class Reference	12
3.12	MPIInput Class Reference	12
3.12.1	Constructor & Destructor Documentation	12
3.12.1.1	MPIInput() [1/2]	12
3.12.1.2	MPIInput() [2/2]	12

3.12.2 Member Function Documentation	13
3.12.2.1 MPIStart()	13
3.13 OMP< T > Class Template Reference	13
3.14 Partstruct Struct Reference	13
3.15 PassFail Class Reference	14
3.16 part1::Point Class Reference	14
3.16.1 Detailed Description	15
3.17 Progression Class Reference	15
3.18 QTstyle_Test Class Reference	16
3.18.1 Detailed Description	16
3.19 RandomNumberGenerator Class Reference	16
3.20 Stack< T, CONT > Class Template Reference	17
3.21 StandardNormalDistribution Class Reference	17
3.21.1 Detailed Description	18
3.22 StatisticalDistribution Class Reference	18
3.22.1 Detailed Description	19
3.22.2 Constructor & Destructor Documentation	19
3.22.2.1 StatisticalDistribution()	20
3.22.2.2 ~StatisticalDistribution()	20
3.23 Str Class Reference	20
3.23.1 Detailed Description	21
3.23.2 Constructor & Destructor Documentation	21
3.23.2.1 Str() [1/2]	21
3.23.2.2 Str() [2/2]	21
3.24 Student_info Class Reference	22
3.25 SYN_Mat< T > Class Template Reference	22
3.26 TemplateUnderTest< T > Class Template Reference	23
3.27 Trap Class Reference	23
3.28 tutorial::Vec< T > Class Template Reference	24
3.29 Vec< T > Class Template Reference	24
3.30 VectorMethodTest Class Reference	25

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Genfun::Argument	5
background_task	7
Complex	7
Core	8
Audit	6
Grad	9
PassFail	14
InitiateVectorMethod< ItemType >	10
MPI_BC_Generic< T, Q, R >	11
MPI_sorting_methods	12
MPIInput	12
OMP< T >	13
Partstruct	13
part1::Point	14
Progression	15
ArithProgression	5
QTeststyle_Test	16
RandomNumberGenerator	16
Stack< T, CONT >	17
StatisticalDistribution	18
StandardNormalDistribution	17
Str	20
Student_info	22
TemplateUnderTest< T >	23
TestCase	
MPI_BC	10
SYN_Mat< T >	22
Vec< T >	24
Vec< char >	24
TestFixture	
VectorMethodTest	25
Trap	23
tutorial::Vec< T >	24

Chapter 2

Class Index

2.1 Class List

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

Genfun::Argument	5
ArithProgression	5
Audit	6
background_task	7
Complex	7
Core	8
Grad	9
InitiateVectorMethod< ItemType >	10
MPI_BC	10
MPI_BC_Generic< T, Q, R >	11
MPI_sorting_methods	12
MPIInput	12
OMP< T >	13
Partstruct	13
PassFail	14
part1::Point	14
Progression	15
QTstyle_Test	
A test class - find and replace from this template for future class definitions	16
RandomNumberGenerator	16
Stack< T, CONT >	17
StandardNormalDistribution	
Standard Normal Distribution Implementation	17
StatisticalDistribution	
Statistical Distribution Class	18
Str	
A constructor with a character pointer input	20
Student_info	22
SYN_Mat< T >	22
TemplateUnderTest< T >	23
Trap	23
tutorial::Vec< T >	24
Vec< T >	24
VectorMethodTest	25

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

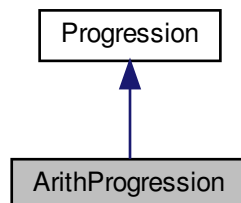
- std::ostream & **operator<<** (std::ostream &o, const [Argument](#) &a)

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

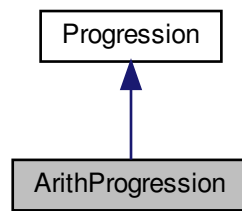
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Argument.h

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

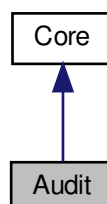
- long **inc**

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

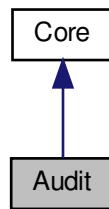
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

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

Additional Inherited Members

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

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

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 Complex Class Reference

Public Member Functions

- **Complex** (double r, double i=0)

Friends

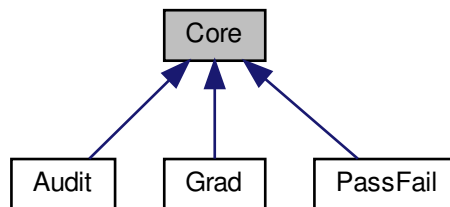
- `bool operator== (const Complex &a, const Complex &b)`

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

- `/home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp2.cxx`

3.6 Core Class Reference

Inheritance 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`

Protected Member Functions

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

Protected Attributes

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

Friends

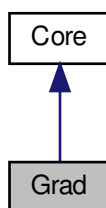
- class **Student_info**

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

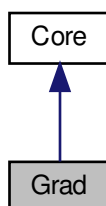
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp_dynamicbindingandinheritance.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp_dynamicbindingandinheritance.cxx

3.7 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

Additional Inherited Members

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

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

3.8 InitiateVectorMethod< ItemType > Class Template Reference

Public Member Functions

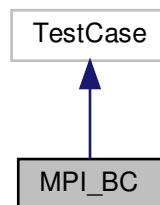
- **InitiateVectorMethod** (int, int)
- void **setup** (int *)
- void **traits** ()
- void **SendVector** ()
- void **GetData** ()

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

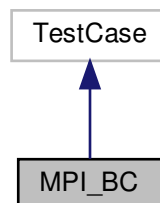
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/new.hpp

3.9 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 *)

3.9.1 Constructor & Destructor Documentation

3.9.1.1 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

3.10 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

3.11 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

3.12 MPIInput Class Reference

Public Member Functions

- [MPIInput](#) ()
[MPIInput](#) class - constructor.
- [MPIInput](#) (int, int)
[MPIInput](#) class - constructor.
- void [MPIStart](#) ()
[MPIInput](#) class - MPIStart method.
- void **getData** ()
- void **bubbleSort** ()
- void **oddEvenSort** ()
- void **I_send** ()

3.12.1 Constructor & Destructor Documentation

3.12.1.1 MPIInput() [1/2]

```
MPIInput::MPIInput ( )
```

[MPIInput](#) class - constructor.

Default constructor

The default constructor

3.12.1.2 MPIInput() [2/2]

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

[MPIInput](#) class - constructor.

int int constructor

The input constructor

3.12.2 Member Function Documentation

3.12.2.1 MPIStart()

```
void MPIInput::MPIStart ( )
```

[MPIInput](#) class - MPIStart method.

Sets up the processor ranks and size for use later - could really be incorporated into the constructor

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

3.13 OMP< T > Class Template Reference

Public Member Functions

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

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

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

3.14 Partstruct Struct Reference

Public Attributes

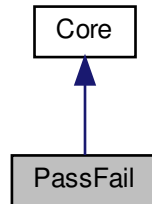
- int **class**
- double **d** [6]
- char **b** [7]

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

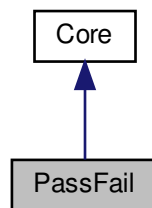
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI_struct.cxx

3.15 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

Additional Inherited Members

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

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

3.16 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**

3.16.1 Detailed Description

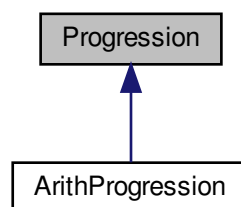
This is a simple 3D point class

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

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

3.17 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 quantitative finance is that of a statistical distribution. Random variables play a huge part in quantitative financial modelling. Derivatives, pricing, cash-flow forecasting and quantitative trading all make use of statistical 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 distributions such that we can use the static classes elsewhere without bringing along the heavy random number generation functions.

Equally useful is the fact that we will be able to "swap out" different random number generators for our statistics 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:

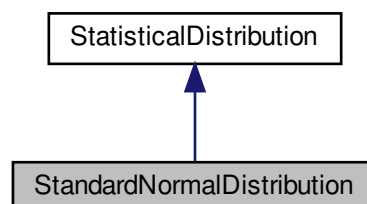
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp_templates.hpp

3.21 StandardNormalDistribution Class Reference

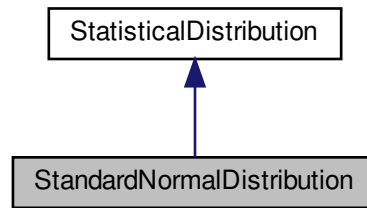
Standard Normal Distribution Implementation.

```
#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:

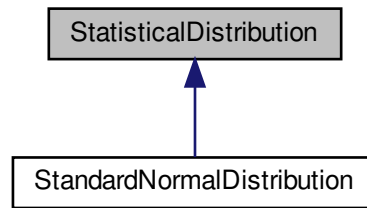
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/statistics.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/statistics.cxx

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
Variable 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**
- typedef [Vec](#)< char >::size_type **size_type**

Public Member Functions

- **Str** (size_type n, char c)
- **Str** (const char *cp)
- template<class In >
 Str (In b, In e)
- [Str](#) ()
- **Str** (size_type n, char c)
- [Str](#) (const char *cp)
 A constructor with a character pointer input.
- 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 sequence 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 independent.

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 can 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 can 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 approximately as we would like.

3.23.2 Constructor & Destructor Documentation

3.23.2.1 [Str\(\)](#) [1/2]

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

default constructor, create an empty str

3.23.2.2 [Str\(\)](#) [2/2]

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

A constructor with a character pointer input.

create a [Str](#) containing n copies of c

Copy constructor

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

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp2.cxx
- /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

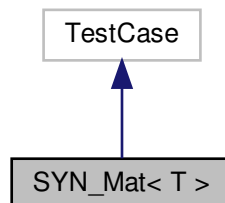
- static bool **compare** (const [Student_info](#) &s1, const [Student_info](#) &s2)

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

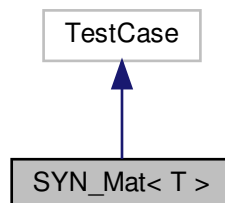
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Student_info.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/Student_info.cxx

3.25 SYN_Mat< T > Class Template Reference

Inheritance diagram for SYN_Mat< T >:



Collaboration diagram for SYN_Mat< T >:



Public Member Functions

- **SYN_Mat** (unsigned _rows, unsigned _cols, const T &_initial)
- **SYN_Mat** (const [SYN_Mat](#)< T > &alloc)
- [SYN_Mat](#)< T > & **operator=** (const [SYN_Mat](#)< T > &alloc)
- [SYN_Mat](#)< T > **operator+** (const [SYN_Mat](#)< T > &rhs)
- [SYN_Mat](#)< T > & **operator+=** (const [SYN_Mat](#)< T > &rhs)
- [SYN_Mat](#)< T > **operator-** (const [SYN_Mat](#)< T > &rhs)
- [SYN_Mat](#)< T > & **operator-=** (const [SYN_Mat](#)< T > &rhs)
- [SYN_Mat](#)< T > **operator*** (const [SYN_Mat](#)< T > &rhs)
- [SYN_Mat](#)< T > & **operator*=** (const [SYN_Mat](#)< T > &rhs)
- [SYN_Mat](#)< T > **transpose** ()
- [SYN_Mat](#)< T > **operator+** (const T &rhs)
- [SYN_Mat](#)< T > **operator-** (const T &rhs)
- [SYN_Mat](#)< T > **operator*** (const T &rhs)
- [SYN_Mat](#)< T > **operator/** (const T &rhs)
- `std::vector< T >` **operator*** (const `std::vector< T >` &rhs)
- `std::vector< T >` **diag_vec** ()
- T & **operator()** (const unsigned &row, const unsigned &col)
- const T & **operator()** (const unsigned &row, const unsigned &col) const
- unsigned **get_rows** () const
- unsigned **get_cols** () const
- void **test1** ()
- void **test2** ()

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

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

3.26 `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

3.27 `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

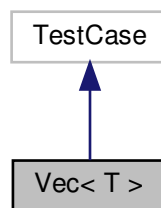
3.28 tutorial::Vec< T > Class Template Reference

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

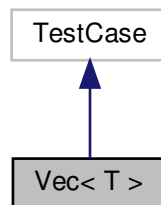
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/lib_mpi.hpp

3.29 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

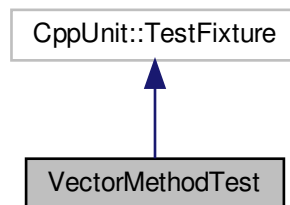
- **Vec** (size_type n, const T &t=T())
- **Vec** (const **Vec** &v)
- **Vec** & **operator=** (const **Vec** &)
- const T & **operator[]** (size_type i) const
- void **push_back** (const T &t)
- size_type **size** () const
- iterator **begin** ()
- const_iterator **begin** () const
- iterator **end**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

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

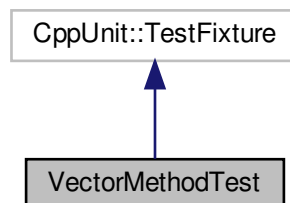
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI_str.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp1.hpp

3.30 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

- ~StatisticalDistribution
 - StatisticalDistribution, [20](#)
- ArithProgression, [5](#)
- Audit, [6](#)
- background_task, [7](#)
- Complex, [7](#)
- Core, [8](#)
- Genfun::Argument, [5](#)
- Grad, [9](#)
- InitiateVectorMethod< ItemType >, [10](#)
- MPI_BC_Generic< T, Q, R >, [11](#)
- MPI_BC, [10](#)
 - MPI_BC, [11](#)
- MPI_sorting_methods, [12](#)
- MPIInput, [12](#)
 - MPIInput, [12](#)
 - MPIStart, [13](#)
- MPIStart
 - MPIInput, [13](#)
- OMP< T >, [13](#)
- part1::Point, [14](#)
- Partstruct, [13](#)
- PassFail, [14](#)
- Progression, [15](#)
- QTstyle_Test, [16](#)
- RandomNumberGenerator, [16](#)
- SYN_Mat< T >, [22](#)
- Stack< T, CONT >, [17](#)
- StandardNormalDistribution, [17](#)
- StatisticalDistribution, [18](#)
 - ~StatisticalDistribution, [20](#)
 - StatisticalDistribution, [19](#)
- Str, [20](#)
 - Str, [21](#)
- Student_info, [22](#)
- TemplateUnderTest< T >, [23](#)
- Trap, [23](#)
- tutorial::Vec< T >, [24](#)
- Vec< T >, [24](#)
- VectorMethodTest, [25](#)