libAHmath Reference Manual

0.1

Generated by Doxygen 1.4.4

Fri Apr 7 01:18:17 2006

Contents

Chapter 1

libAHmath Directory Hierarchy

1.1 libAHmath Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

Chapter 2

libAHmath Namespace Index

2.1 libAHmath Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

libAHmath Hierarchical Index

3.1 libAHmath Class Hierarchy

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

Chapter 4

libAHmath Class Index

4.1 libAHmath Class List

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

Chapter 5

libAHmath File Index

5.1 libAHmath File List

Here is a list of all files with brief descriptions:

Chapter 6

libAHmath Directory Documentation

6.1 AH/ Directory Reference

Directories

- directory Math

6.2 AH/Math/Ft/ Directory Reference

Files

- file Wisdom.h

6.3 AH/Math/ Directory Reference

Directories

- directory Ft

Files

- file DataConv.h
- file DataStat.h
- file Dither.h
- file NullSearch.h
- file NullSearchCreator.h
- file ReduceSpectrum.h
- file RegulaFalsi.h

Chapter 7

libAHmath Namespace Documentation

7.1 Ah Namespace Reference

Classes

- class NullSearch

  represents a generic search class for zero crossing points

- class NullSearchCreator

  represents the NullSearch creator class

- class RegulaFalsi

  represents a regula falsi search class for zero crossing points

7.2 AH Namespace Reference

Classes

- class DataConv
- class DataStat
- class ReduceSpectrum

Chapter 8

libAHmath Class Documentation

8.1 AH::DataConv Class Reference

`#include <DataConv.h>`

Static Public Member Functions

- static void reduceLinear (const double *data, unsigned int dataNum, unsigned int outNum, double *outMean, double *outMax, double *outMin)

  reduce data linear

8.1.1 Member Function Documentation

8.1.1.1 static void AH::DataConv::reduceLinear (const double * *data*, unsigned int *dataNum*, unsigned int *outNum*, double * *outMean*, double * *outMax*, double * *outMin*) [static]

reduce data linear

DataConv::reduceLinear() takes *dataNum* double values from *data* and reduces them to *outNum* double values in one or more of *outMean*, *outMax* and *outMin*, if these are not NULL pointers. Usually *outNum*> is smaller than *dataNum*, so

- *outMean* contains the mean value of those data points in *data* which correspond to one point in *outMean*.
- *outMax* contains the maximum value of those data points in *data* which correspond to one point in *outMax*.
- *outMin* contains the minimum value of those data points in *data* which correspond to one point in *outMin*.

If *outNum*> is **not**smaller than *dataNum*, then *outMean*, *outMax* and *outMin* contain the same values.

Parameters:

data input data

dataNum number of input data

outNum number of output data

outMean output data for mean value, may be 0 if not needed

outMax output data for maximum value, may be 0 if not needed

outMin output data for minimum value, may be 0 if not needed

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

- AH/Math/DataConv.h

8.2 AH::DataStat Class Reference

`#include <DataStat.h>`

Static Public Member Functions

- static void calcMean (const double *data, unsigned int dataNum, double *outMean)

  calculate mean value of data array

- static void calcMaxMin (const double *data, unsigned int dataNum, double *outMax, double *outMin)

  calculate max and minimum value of data array

- static void calcMeanMaxMin (const double *data, unsigned int dataNum, double *outMean, double *outMax, double *outMin)

  calculate mean, max and minimum value of data array

8.2.1 Member Function Documentation

8.2.1.1 static void AH::DataStat::calcMaxMin (const double * *data*, unsigned int *dataNum*, double * *outMax*, double * *outMin*) [static]

calculate max and minimum value of data array

DataStat::calcMaxMin() calculates *dataNum* double values from *data* and stores the maximum value in *outMax* and the minimum value in *outMin*, if these are not NULL pointers.

Parameters:

data input data

dataNum number of input data

outMax for maximum value, may be 0 if not needed

outMin for minimum value, may be 0 if not needed

8.2.1.2 static void AH::DataStat::calcMean (const double * *data*, unsigned int *dataNum*, double * *outMean*) [static]

calculate mean value of data array

DataStat::calcMean() calculates *dataNum* double values from *data* and stores the mean value in *outMean*.

Parameters:

data input data

dataNum number of input data

outMean for mean value

8.2.1.3 static void AH::DataStat::calcMeanMaxMin (const double * *data*, unsigned int *dataNum*, double * *outMean*, double * *outMax*, double * *outMin*) [static]

calculate mean, max and minimum value of data array

DataStat::calcMeanMaxMin() calculates *dataNum* double values from *data* and stores the mean value in *outMean*, the maximum value in *outMax* and the minimum value in *outMin*, if these are not NULL pointers.

Parameters:

data input data

dataNum number of input data

outMean for mean value, may be 0 if not needed

outMax for maximum value, may be 0 if not needed

outMin for minimum value, may be 0 if not needed

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

- AH/Math/DataStat.h

8.3 MathDither Class Reference

`#include <Dither.h>`

Public Types

- enum Type { None, RectAngle, Triangle }

Public Member Functions

- MathDither (Type t, unsigned int seed=1)
- ~MathDither ()
- double getSignal ()

Static Public Member Functions

- static double getRectAngle ()

8.3.1 Member Enumeration Documentation

8.3.1.1 enum MathDither::Type

Enumerator:

None

RectAngle

Triangle

8.3.2 Constructor & Destructor Documentation

8.3.2.1 MathDither::MathDither (Type *t*, unsigned int *seed* = 1)

8.3.2.2 MathDither::~MathDither () [inline]

8.3.3 Member Function Documentation

8.3.3.1 static double MathDither::getRectAngle () [static]

8.3.3.2 double MathDither::getSignal ()

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

- AH/Math/Dither.h

8.4 MathFtWisdom Class Reference

`#include <Wisdom.h>`

Public Member Functions

- MathFtWisdom (unsigned int debug=0)
- virtual ~MathFtWisdom ()
- rfftw_plan getPlan (unsigned int num, fftw_direction direction=FFTW_FORWARD)

Classes

- struct **Entry**

8.4.1 Constructor & Destructor Documentation

8.4.1.1 MathFtWisdom::MathFtWisdom (unsigned int *debug* = 0)

8.4.1.2 virtual MathFtWisdom::~MathFtWisdom () [virtual]

8.4.2 Member Function Documentation

8.4.2.1 rfftw_plan MathFtWisdom::getPlan (unsigned int *num*, fftw_direction *direction* = FFTW_FORWARD)

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

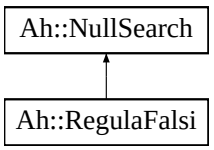
- AH/Math/Ft/Wisdom.h

8.5 Ah::NullSearch Class Reference

represents a generic search class for zero crossing points

`#include <NullSearch.h>`

Inheritance diagram for Ah::NullSearch::



Public Member Functions

- NullSearch (double(∗searchFunc)(double))
- virtual ∼NullSearch ()
- virtual double search (double lowLim, double hiLim, double eps)=0

  search the zero crossing point

Protected Attributes

- double(∗ f)(double)

8.5.1 Detailed Description

represents a generic search class for zero crossing points

{ NullSearch} is a pure virtual zero crossing search base class, which just defines the common interface to all search classes.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 Ah::NullSearch::NullSearch (double(∗)(double) *searchFunc*) `[inline]`

8.5.2.2 virtual Ah::NullSearch::∼NullSearch () `[inline, virtual]`

8.5.3 Member Function Documentation

8.5.3.1 virtual double Ah::NullSearch::search (double *lowLim*, double *hiLim*, double *eps*) `[pure virtual]`

search the zero crossing point

Implemented in Ah::RegulaFalsi.

8.5.4 Member Data Documentation

8.5.4.1 double(* Ah::NullSearch::f)(double) [protected]

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

- AH/Math/NullSearch.h

8.6 Ah::NullSearchCreator Class Reference

represents the NullSearch creator class

`#include <NullSearchCreator.h>`

Public Member Functions

- NullSearchCreator ()
  default constructor
- virtual ~NullSearchCreator ()
  virtual destructor
- virtual NullSearch * create (double(*searchFunc)(double), const char *algo=0) const
  create instance of specified nullsearch

8.6.1 Detailed Description

represents the NullSearch creator class

{ NullSearchCreator} represents a creator class for all zero crossing search algorithms within {*libMath}*.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 Ah::NullSearchCreator::NullSearchCreator () [inline]

default constructor

8.6.2.2 virtual Ah::NullSearchCreator::~NullSearchCreator () [inline, virtual]

virtual destructor

8.6.3 Member Function Documentation

8.6.3.1 virtual NullSearch* Ah::NullSearchCreator::create (double(*)(double) *searchFunc*, const char * *algo* = 0) const [virtual]

create instance of specified nullsearch

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

- AH/Math/NullSearchCreator.h

8.7 AH::ReduceSpectrum Class Reference

`#include <ReduceSpectrum.h>`

Public Member Functions

- ReduceSpectrum (int v, unsigned int osize, double fmax, double flo, double fhi)
- void processLinXScale (const double *data, unsigned int isize, double *avout, double *pkout)
- void processLogXScale (const double *data, unsigned int isize, double *avout, double *pkout, double smooth)

8.7.1 Constructor & Destructor Documentation

8.7.1.1 AH::ReduceSpectrum::ReduceSpectrum (int *v*, unsigned int *osize*, double *fmax*, double *flo*, double *fhi*)

8.7.2 Member Function Documentation

8.7.2.1 void AH::ReduceSpectrum::processLinXScale (const double * *data*, unsigned int *isize*, double * *avout*, double * *pkout*)

8.7.2.2 void AH::ReduceSpectrum::processLogXScale (const double * *data*, unsigned int *isize*, double * *avout*, double * *pkout*, double *smooth*)

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

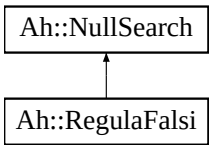
- AH/Math/ReduceSpectrum.h

8.8 Ah::RegulaFalsi Class Reference

represents a regula falsi search class for zero crossing points

`#include <RegulaFalsi.h>`

Inheritance diagram for Ah::RegulaFalsi::



Public Member Functions

- RegulaFalsi (double(∗searchFunc)(double))
- virtual double search (double lowLim, double hiLim, double eps)

  search the zero crossing point

8.8.1 Detailed Description

represents a regula falsi search class for zero crossing points

{ RegulaFalsi} is a zero crossing search base class, which uses the {*Regula* Falsi} algorithm.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 Ah::RegulaFalsi::RegulaFalsi (double(∗)(double) *searchFunc*) `[inline]`

8.8.3 Member Function Documentation

8.8.3.1 virtual double Ah::RegulaFalsi::search (double *lowLim*, double *hiLim*, double *eps*) `[virtual]`

search the zero crossing point

Implements Ah::NullSearch.

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

- AH/Math/RegulaFalsi.h

Chapter 9

libAHmath File Documentation

9.1 AH/Math/DataConv.h File Reference

```
#include <AH/Config/Platform.h>
```

Namespaces

- namespace AH

Classes

- class AH::DataConv

9.2 AH/Math/DataStat.h File Reference

```
#include <AH/Config/Platform.h>
```

Namespaces

- namespace AH

Classes

- class AH::DataStat

9.3 AH/Math/Dither.h File Reference

```
#include <string>
#include <drfftw.h>
```

Classes

- class MathDither

9.4 AH/Math/Ft/Wisdom.h File Reference

```
#include <string>
#include <vector>
#include <drfftw.h>
```

Classes

- class MathFtWisdom
- struct **MathFtWisdom::Entry**

9.5 AH/Math/NullSearch.h File Reference

```
#include <AH/Config/Platform.h>
```

Namespaces

- namespace Ah

Classes

- class Ah::NullSearch

  represents a generic search class for zero crossing points

9.6 AH/Math/NullSearchCreator.h File Reference

```
#include <AH/Config/Platform.h>
```

Namespaces

- namespace Ah

Classes

- class Ah::NullSearchCreator

  represents the NullSearch creator class

9.7 AH/Math/ReduceSpectrum.h File Reference

```
#include <AH/Config/Platform.h>
```

Namespaces

- namespace AH

Classes

- class AH::ReduceSpectrum

9.8 AH/Math/RegulaFalsi.h File Reference

```
#include <AH/Math/NullSearch.h>
```

Namespaces

- namespace Ah

Classes

- class Ah::RegulaFalsi

  represents a regula falsi search class for zero crossing points

Index