DDM-DVS

Generated by Doxygen 1.8.9.1

Tue Sep 29 2015 08:06:05

Contents

Chapter 1

Grupo de Modelación Matemática y Computacional, UNAM

1.1 Introducción

Para compilar el código usar:

```
$ make deps
```

```
$ make
```

Para ejecutar el código usar:

```
$ make run
```

Chapter 2

Todo List

Class ErrorControl

Exception handling

Class MatrizDispersa

Hacer comportamiento para cambiar tamano de banda

Multiplicacion de matrices

Chapter 3

Bug List

Class DPMainMPI
: No hay errores conocidos

Class ErrorControl
: No errors detected

Class EsquemaMEMPI
: No hay errores conocidos

Class ICGM
: No hay errores conocidos

Class IDQGMRES
: No hay errores conocidos

Class LM1MPI
: No hay errores conocidos

Class LM2MPI
: No hay errores conocidos

Class Matriz_Base
: No hay errores conocidos

Class MatrizDispersa
: No hay errores conocidos

Class MF1MPI
: No hay errores conocidos

Class MF2MPI
: No hay errores conocidos

Class PLM1MPI
: No hay errores conocidos

Class PLM2MPI
: No hay errores conocidos

Class PMF1MPI
: No hay errores conocidos

Class PMF2MPI
: No hay errores conocidos

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Class Index

5.1 Class List

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

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

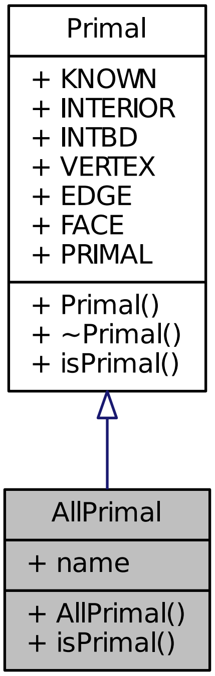
Chapter 7

Class Documentation

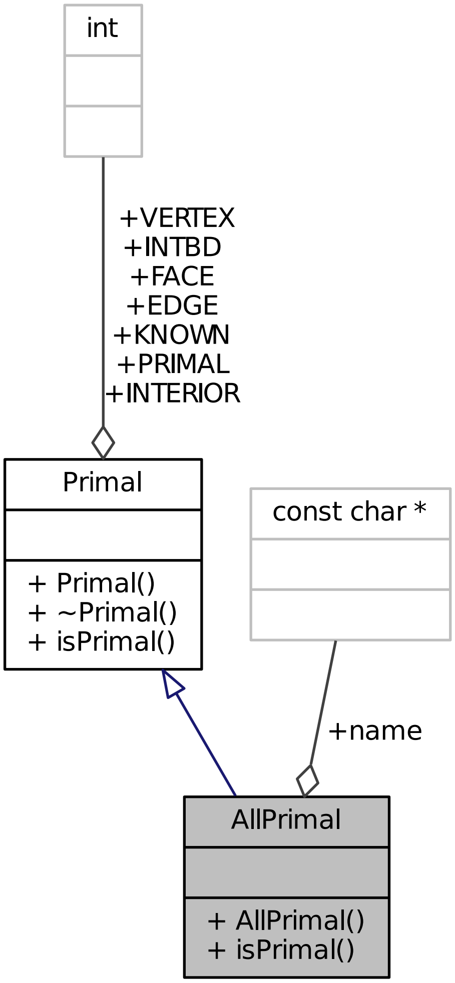
7.1 AllPrimal Class Reference

`#include <AllPrimal.hpp>`

Inheritance diagram for AllPrimal:



Collaboration diagram for AllPrimal:



Public Member Functions

- AllPrimal (void)
- bool isPrimal (int type, int ∗coordN, int ∗coordM)

Public Attributes

- const char ∗ name

Additional Inherited Members

7.1.1 Constructor & Destructor Documentation

7.1.1.1 AllPrimal::AllPrimal (void) [inline]

7.1.2 Member Function Documentation

7.1.2.1 bool AllPrimal::isPrimal (int *type,* int ∗ *coordN,* int ∗ *coordM*) [inline], [virtual]

Implements Primal.

7.1.3 Member Data Documentation

7.1.3.1 const char∗ AllPrimal::name

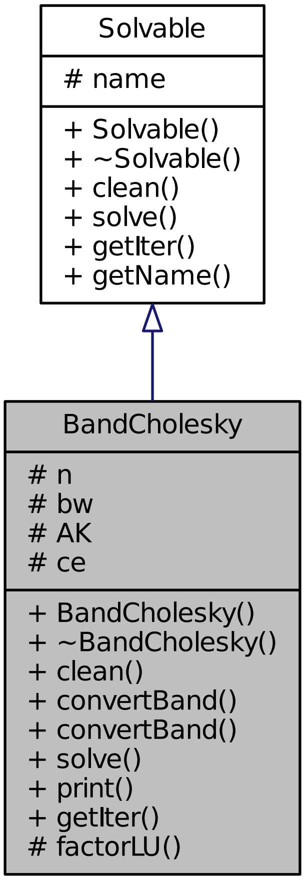
The documentation for this class was generated from the following file:

- AllPrimal.hpp

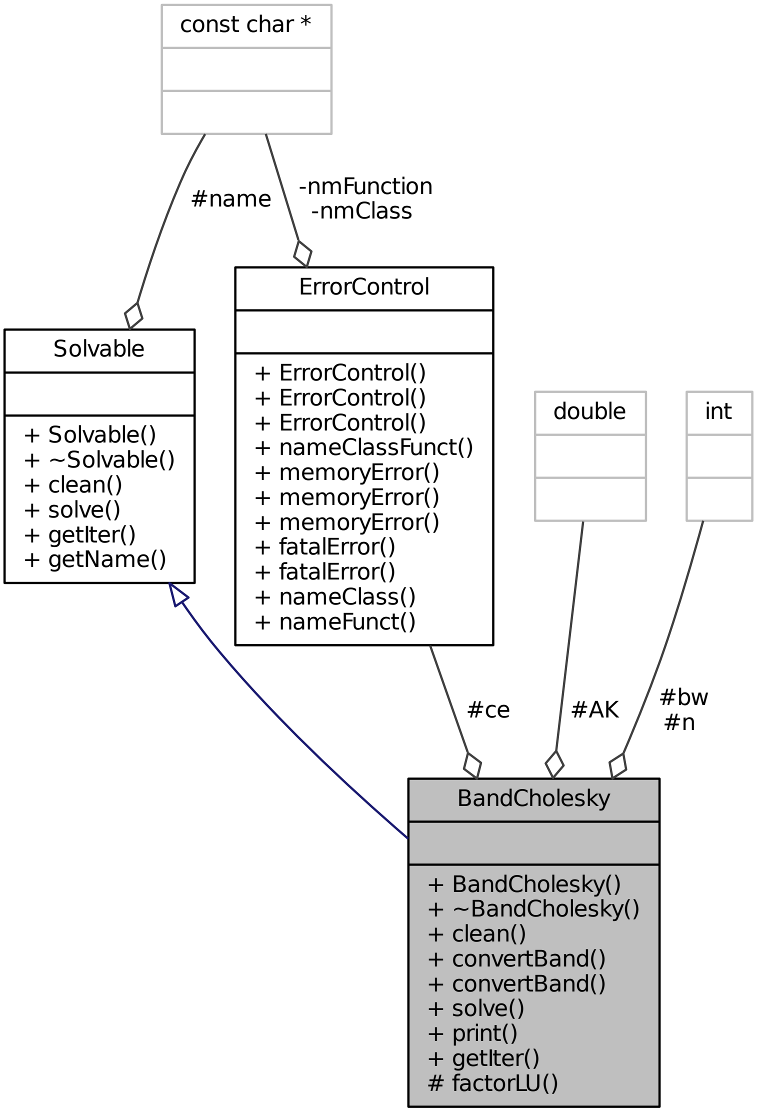
7.2 BandCholesky Class Reference

```
#include <BandCholesky.hpp>
```

Inheritance diagram for BandCholesky:



Collaboration diagram for BandCholesky:



Public Member Functions

- BandCholesky (int n, MatrizDispersa ∗A)
- ∼BandCholesky ()
- void clean (void)
- void convertBand (int n, ldouble ∗∗A)
- void convertBand (int n, MatrizDispersa ∗A)
- void solve (ldouble ∗x, ldouble ∗y)
- void print (void)

- int getIter (void)

Protected Member Functions

- void factorLU (void)

Protected Attributes

- int n
- int bw
- ldouble ∗∗ AK
- ErrorControl ce

  Control de errores.

7.2.1 Constructor & Destructor Documentation

7.2.1.1 BandCholesky::BandCholesky (int *n*, MatrizDispersa ∗ *A*) [inline]

7.2.1.2 BandCholesky::∼BandCholesky ( ) [inline]

7.2.2 Member Function Documentation

7.2.2.1 void BandCholesky::clean (void) [inline], [virtual]

Implements Solvable.

7.2.2.2 void BandCholesky::convertBand (int *n*, ldouble ∗∗ *A*)

7.2.2.3 void BandCholesky::convertBand (int *n*, MatrizDispersa ∗ *A*)

7.2.2.4 void BandCholesky::factorLU (void) [protected]

7.2.2.5 int BandCholesky::getIter (void) [inline], [virtual]

Implements Solvable.

7.2.2.6 void BandCholesky::print (void)

7.2.2.7 void BandCholesky::solve (ldouble ∗ *x*, ldouble ∗ *y*) [virtual]

Implements Solvable.

7.2.3 Member Data Documentation

7.2.3.1 ldouble∗∗ BandCholesky::AK [protected]

7.2.3.2 int BandCholesky::bw [protected]

7.2.3.3 ErrorControl BandCholesky::ce [protected]

Control de errores.

7.2.3.4 int BandCholesky::n [protected]

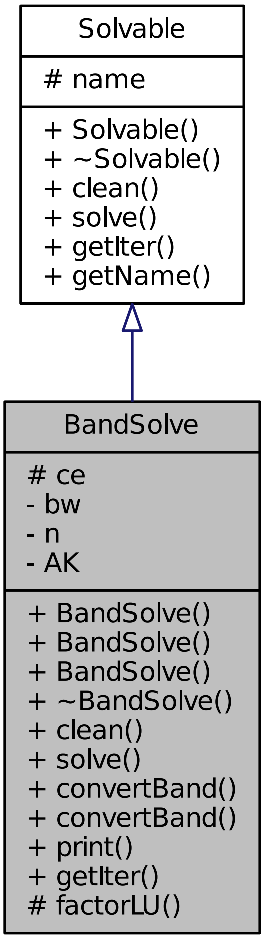
The documentation for this class was generated from the following files:

- BandCholesky.hpp
- BandCholesky.cpp

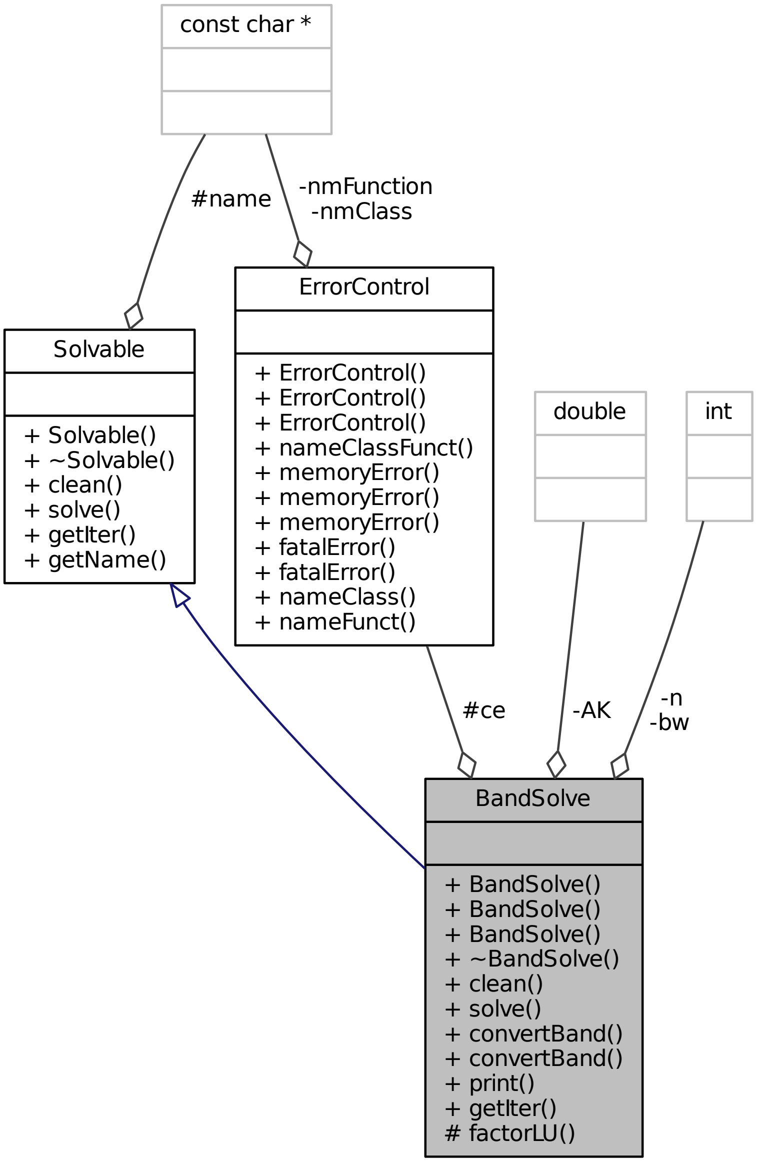
7.3 BandSolve Class Reference

```
#include <BandSolve.hpp>
```

Inheritance diagram for BandSolve:



Collaboration diagram for BandSolve:



Public Member Functions

- BandSolve (void)
- BandSolve (int n, ldouble ∗∗A)
- BandSolve (int n, MatrizDispersa ∗A)
- ∼BandSolve ()
- void clean (void)

- void solve (ldouble ∗x, ldouble ∗y)
- void convertBand (int n, ldouble ∗∗A)
- void convertBand (int n, MatrizDispersa ∗A)
- void print (void)
- int getIter (void)

Protected Member Functions

- void factorLU (void)

Protected Attributes

- ErrorControl ce

Private Attributes

- int bw
- int n
- ldouble ∗∗ AK

7.3.1 Constructor & Destructor Documentation

7.3.1.1 BandSolve::BandSolve (void) [inline]

7.3.1.2 BandSolve::BandSolve (int *n*, ldouble ∗∗ *A*)

7.3.1.3 BandSolve::BandSolve (int *n*, MatrizDispersa ∗ *A*)

7.3.1.4 BandSolve::∼BandSolve ( ) [inline]

7.3.2 Member Function Documentation

7.3.2.1 void BandSolve::clean (void) [inline], [virtual]

Implements Solvable.

7.3.2.2 void BandSolve::convertBand (int *n*, ldouble ∗∗ *A*)

7.3.2.3 void BandSolve::convertBand (int *n*, MatrizDispersa ∗ *A*)

7.3.2.4 void BandSolve::factorLU (void) [protected]

7.3.2.5 int BandSolve::getIter (void) [inline], [virtual]

Implements Solvable.

7.3.2.6 void BandSolve::print (void)

7.3.2.7 void BandSolve::solve (**ldouble** ∗ *x*, **ldouble** ∗ *y*) `[virtual]`

Implements Solvable.

7.3.3 Member Data Documentation

7.3.3.1 **ldouble**∗∗ BandSolve::AK `[private]`

7.3.3.2 int BandSolve::bw `[private]`

7.3.3.3 **ErrorControl** BandSolve::ce `[protected]`

7.3.3.4 int BandSolve::n `[private]`

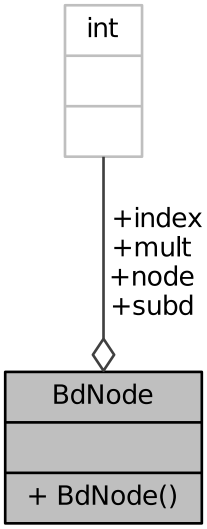
The documentation for this class was generated from the following files:

- BandSolve.hpp
- BandSolve.cpp

7.4 BdNode Class Reference

```
#include <BdNode.hpp>
```

Collaboration diagram for BdNode:



Public Member Functions

- BdNode (int s, int n, int i, int m)

Public Attributes

- int subd
- int node
- int index
- int mult

7.4.1 Constructor & Destructor Documentation

7.4.1.1 BdNode::BdNode (int *s*, int *n*, int *i*, int *m*) `[inline]`

7.4.2 Member Data Documentation

7.4.2.1 int BdNode::index

7.4.2.2 int BdNode::mult

7.4.2.3 int BdNode::node

7.4.2.4 int BdNode::subd

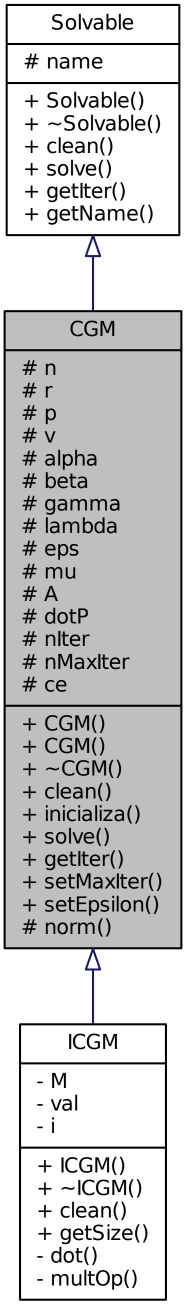
The documentation for this class was generated from the following file:

- BdNode.hpp

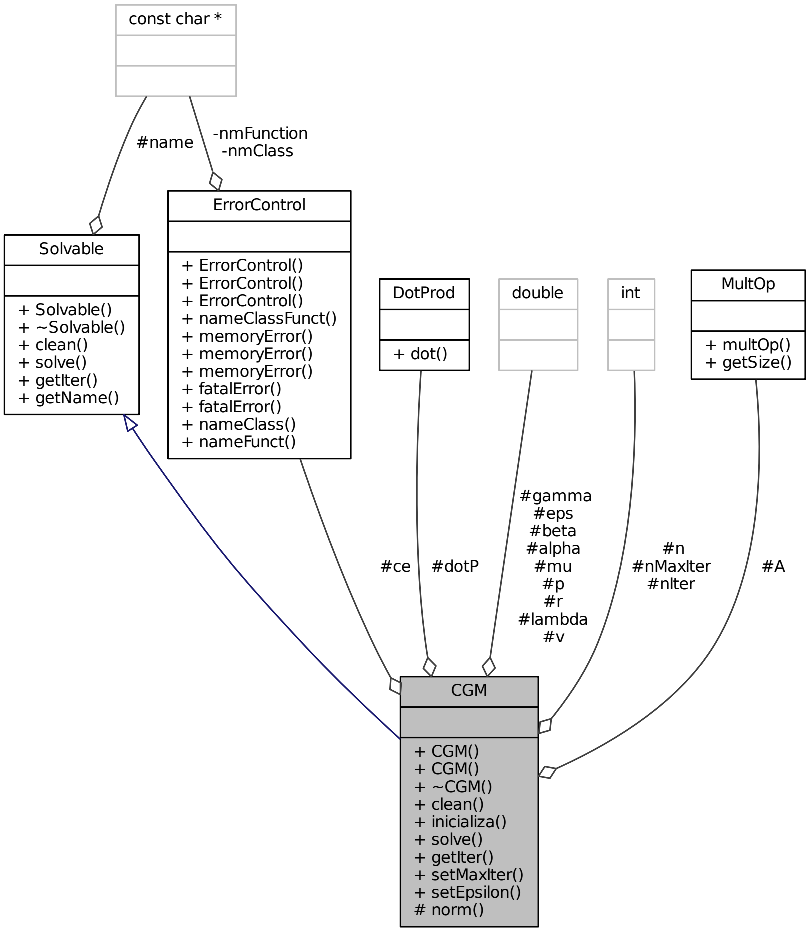
7.5 CGM Class Reference

```
#include <CGM.hpp>
```

Inheritance diagram for CGM:



Collaboration diagram for CGM:



Public Member Functions

- CGM (void)
- CGM (MultOp &A, DotProd &dotP, ldouble eps)
- ∼CGM ()
- void clean (void)
- void inicializa (void)
- void solve (ldouble ∗u, ldouble ∗b)
- int getIter (void)
- void setMaxIter (int nmi)
- void setEpsilon (ldouble ep)

Protected Member Functions

- ldouble norm (ldouble ∗x)

Protected Attributes

- int n
- ldouble ∗ r
- ldouble ∗ p
- ldouble ∗ v
- ldouble alpha
- ldouble beta
- ldouble gamma
- ldouble lambda
- ldouble eps
- ldouble mu
- MultOp ∗ A
- DotProd ∗ dotP
- int nIter
- int nMaxIter
- ErrorControl ce

7.5.1 Constructor & Destructor Documentation

7.5.1.1 CGM::CGM (void) [inline]

7.5.1.2 CGM::CGM (MultOp & *A,* DotProd & *dotP,* ldouble *eps*) [inline]

7.5.1.3 CGM::∼CGM ( ) [inline]

7.5.2 Member Function Documentation

7.5.2.1 void CGM::clean (void) [inline], [virtual]

Implements Solvable.

Reimplemented in ICGM.

7.5.2.2 int CGM::getIter (void) [inline], [virtual]

Implements Solvable.

7.5.2.3 void CGM::inicializa (void) [inline]

7.5.2.4 ldouble CGM::norm (ldouble ∗ *x*) [protected]

7.5.2.5 void CGM::setEpsilon (ldouble *ep*) [inline]

7.5.2.6 void CGM::setMaxIter (int *nmi*) [inline]

7.5.2.7 void CGM::solve (ldouble ∗ *u,* ldouble ∗ *b*) [virtual]

Implements Solvable.

7.5.3 Member Data Documentation

7.5.3.1 MultOp* CGM::A [protected]

7.5.3.2 Idouble CGM::alpha [protected]

7.5.3.3 Idouble CGM::beta [protected]

7.5.3.4 ErrorControl CGM::ce [protected]

7.5.3.5 DotProd* CGM::dotP [protected]

7.5.3.6 Idouble CGM::eps [protected]

7.5.3.7 Idouble CGM::gamma [protected]

7.5.3.8 Idouble CGM::lambda [protected]

7.5.3.9 Idouble CGM::mu [protected]

7.5.3.10 int CGM::n [protected]

7.5.3.11 int CGM::nIter [protected]

7.5.3.12 int CGM::nMaxIter [protected]

7.5.3.13 Idouble * CGM::p [protected]

7.5.3.14 Idouble* CGM::r [protected]

7.5.3.15 Idouble * CGM::v [protected]

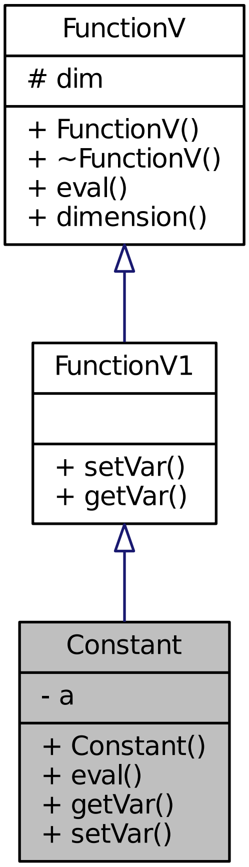
The documentation for this class was generated from the following files:

- CGM.hpp
- CGM.cpp

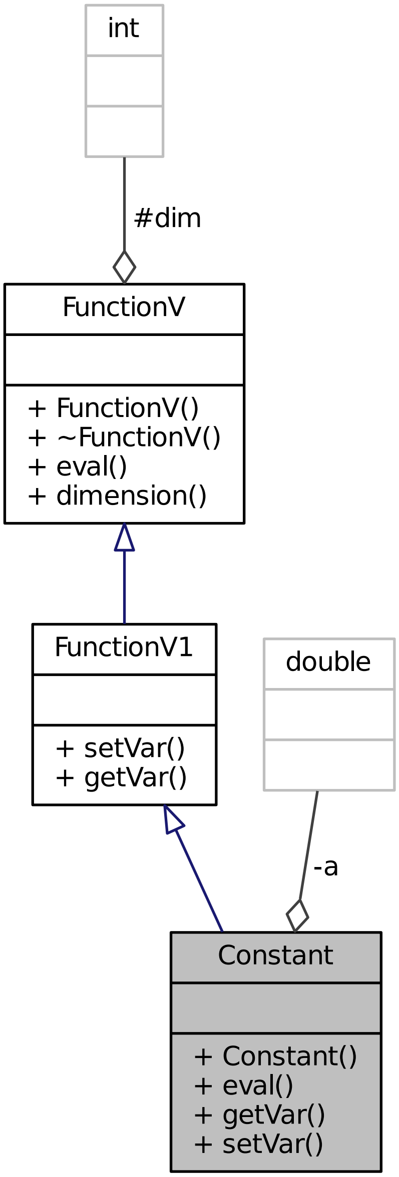
7.6 Constant Class Reference

```
#include <Constant.hpp>
```

Inheritance diagram for Constant:



Collaboration diagram for Constant:



Public Member Functions

- Constant (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- ldouble a

Additional Inherited Members

7.6.1 Constructor & Destructor Documentation

7.6.1.1 Constant::Constant (**ldouble** ***b***) `[inline]`

7.6.2 Member Function Documentation

7.6.2.1 **ldouble** Constant::eval (int ***d,*** **ldouble** ∗ ***x***) `[inline]`, `[virtual]`

Implements FunctionV.

7.6.2.2 **ldouble** Constant::getVar (void) `[inline]`, `[virtual]`

Implements FunctionV1.

7.6.2.3 void Constant::setVar (**ldouble** ***b***) `[inline]`, `[virtual]`

Implements FunctionV1.

7.6.3 Member Data Documentation

7.6.3.1 **ldouble** Constant::a `[private]`

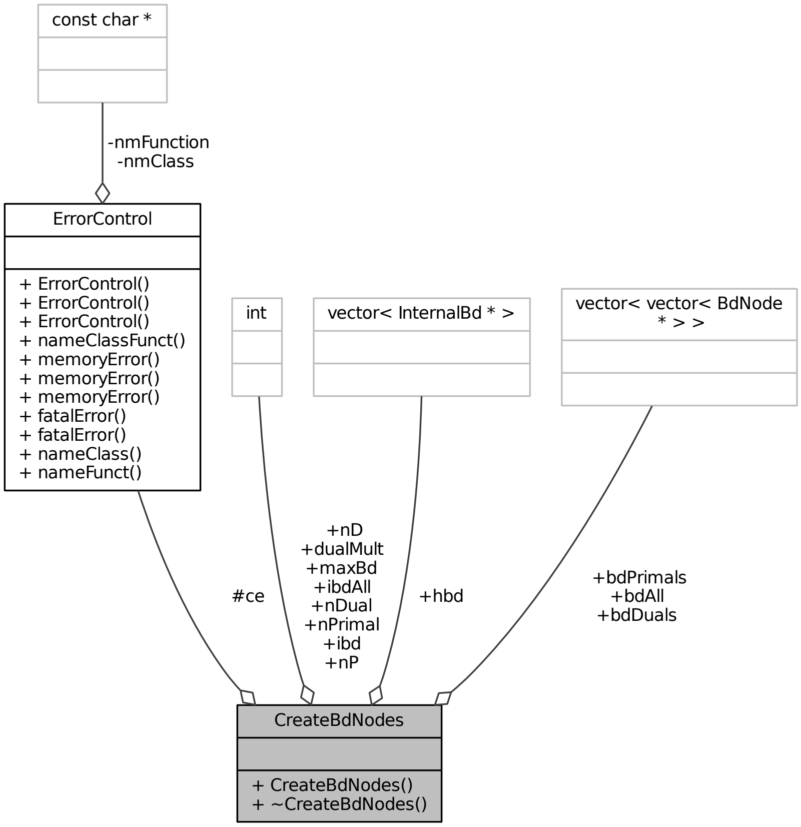
The documentation for this class was generated from the following file:

- Constant.hpp

7.7 CreateBdNodes Class Reference

```
#include <CreateBdNodes.hpp>
```

Collaboration diagram for CreateBdNodes:



Public Member Functions

- CreateBdNodes (void)
- ∼CreateBdNodes ()

Public Attributes

- int nD
- int nP
- int nDual
- int nPrimal
- int maxBd
- int ibd
- int ibdAll
- int ∗ dualMult
- vector< vector< BdNode ∗ > > bdAll
- vector< vector< BdNode ∗ > > bdDuals
- vector< vector< BdNode ∗ > > bdPrimals
- vector< InternalBd ∗ > hbd

Protected Attributes

- ErrorControl ce

  Control de errores.

7.7.1 Constructor & Destructor Documentation

7.7.1.1 CreateBdNodes::CreateBdNodes (void) [inline]

7.7.1.2 CreateBdNodes::∼CreateBdNodes ( ) [inline]

7.7.2 Member Data Documentation

7.7.2.1 vector<vector<BdNode∗> > CreateBdNodes::bdAll

7.7.2.2 vector<vector<BdNode∗> > CreateBdNodes::bdDuals

7.7.2.3 vector<vector<BdNode∗> > CreateBdNodes::bdPrimals

7.7.2.4 ErrorControl CreateBdNodes::ce [protected]

Control de errores.

7.7.2.5 int∗ CreateBdNodes::dualMult

7.7.2.6 vector<InternalBd∗> CreateBdNodes::hbd

7.7.2.7 int CreateBdNodes::ibd

7.7.2.8 int CreateBdNodes::ibdAll

7.7.2.9 int CreateBdNodes::maxBd

7.7.2.10 int CreateBdNodes::nD

7.7.2.11 int CreateBdNodes::nDual

7.7.2.12 int CreateBdNodes::nP

7.7.2.13 int CreateBdNodes::nPrimal

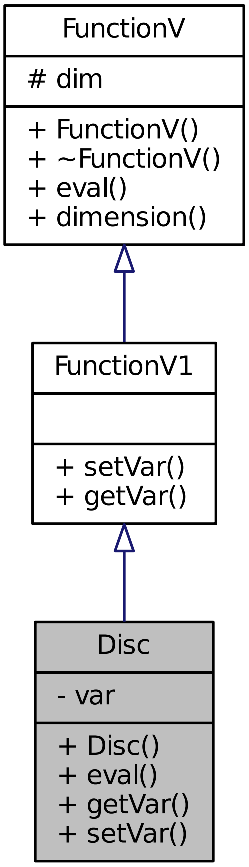
The documentation for this class was generated from the following file:

- CreateBdNodes.hpp

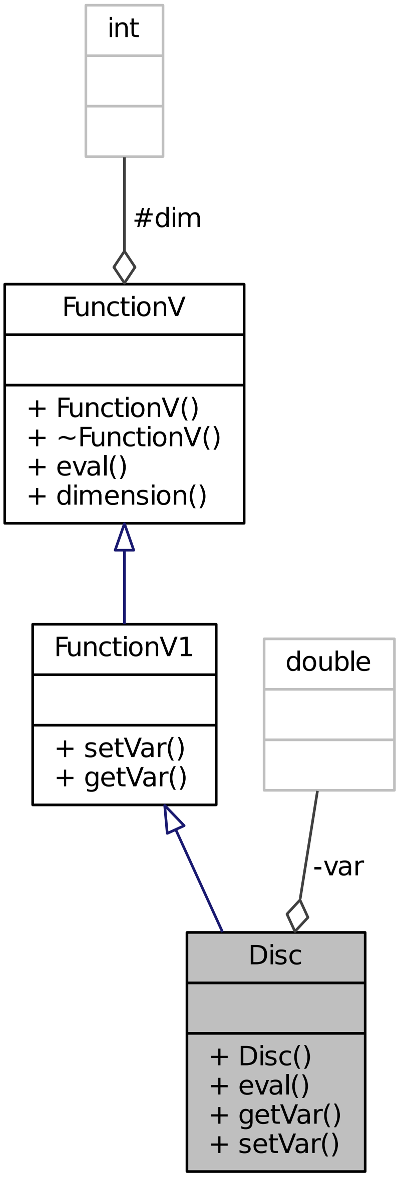
7.8 Disc Class Reference

```
#include <Disc.hpp>
```

Inheritance diagram for Disc:



Collaboration diagram for Disc:



Public Member Functions

- Disc (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.8.1 Constructor & Destructor Documentation

7.8.1.1 Disc::Disc (**Idouble** *b*) `[inline]`

7.8.2 Member Function Documentation

7.8.2.1 **Idouble** Disc::eval (int *d*, **Idouble** ∗ *x*) `[inline], [virtual]`

Implements FunctionV.

7.8.2.2 **Idouble** Disc::getVar (void) `[inline], [virtual]`

Implements FunctionV1.

7.8.2.3 void Disc::setVar (**Idouble** *b*) `[inline], [virtual]`

Implements FunctionV1.

7.8.3 Member Data Documentation

7.8.3.1 **Idouble** Disc::var `[private]`

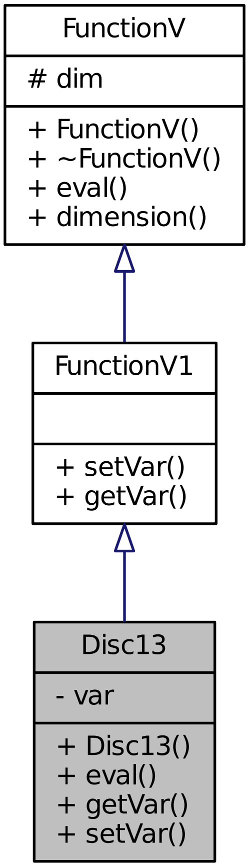
The documentation for this class was generated from the following file:

- Disc.hpp

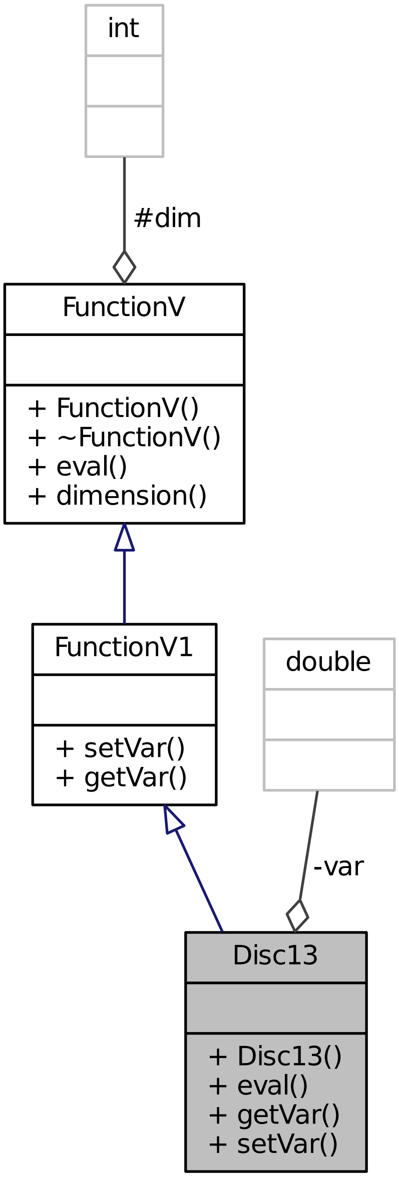
7.9 Disc13 Class Reference

```
#include <Disc13.hpp>
```

Inheritance diagram for Disc13:



Collaboration diagram for Disc13:



Public Member Functions

- Disc13 (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.9.1 Constructor & Destructor Documentation

7.9.1.1 Disc13::Disc13 (Idouble *b*) `[inline]`

7.9.2 Member Function Documentation

7.9.2.1 Idouble Disc13::eval (int *d,* Idouble ∗ *x*) `[inline], [virtual]`

Implements FunctionV.

7.9.2.2 Idouble Disc13::getVar (void) `[inline], [virtual]`

Implements FunctionV1.

7.9.2.3 void Disc13::setVar (Idouble *b*) `[inline], [virtual]`

Implements FunctionV1.

7.9.3 Member Data Documentation

7.9.3.1 Idouble Disc13::var `[private]`

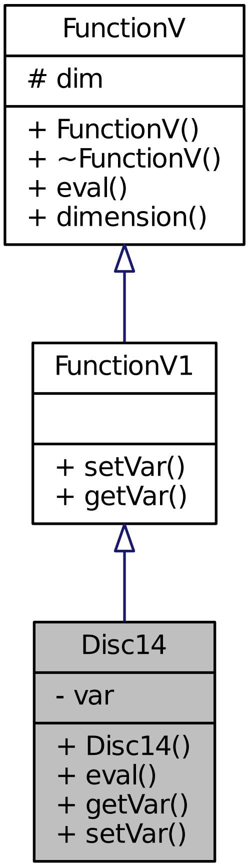
The documentation for this class was generated from the following file:

- Disc13.hpp

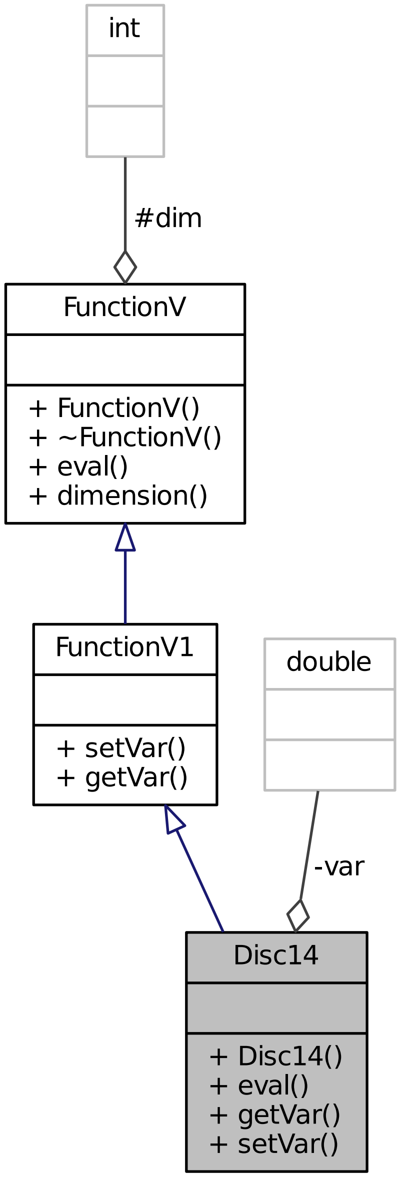
7.10 Disc14 Class Reference

```
#include <Disc14.hpp>
```

Inheritance diagram for Disc14:



Collaboration diagram for Disc14:



Public Member Functions

- Disc14 (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.10.1 Constructor & Destructor Documentation

7.10.1.1 Disc14::Disc14 (Idouble *b*) [inline]

7.10.2 Member Function Documentation

7.10.2.1 Idouble Disc14::eval (int *d,* Idouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.10.2.2 Idouble Disc14::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.10.2.3 void Disc14::setVar (Idouble *b*) [inline], [virtual]

Implements FunctionV1.

7.10.3 Member Data Documentation

7.10.3.1 Idouble Disc14::var [private]

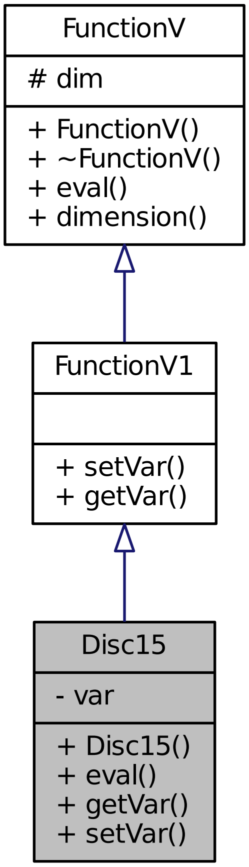
The documentation for this class was generated from the following file:

- Disc14.hpp

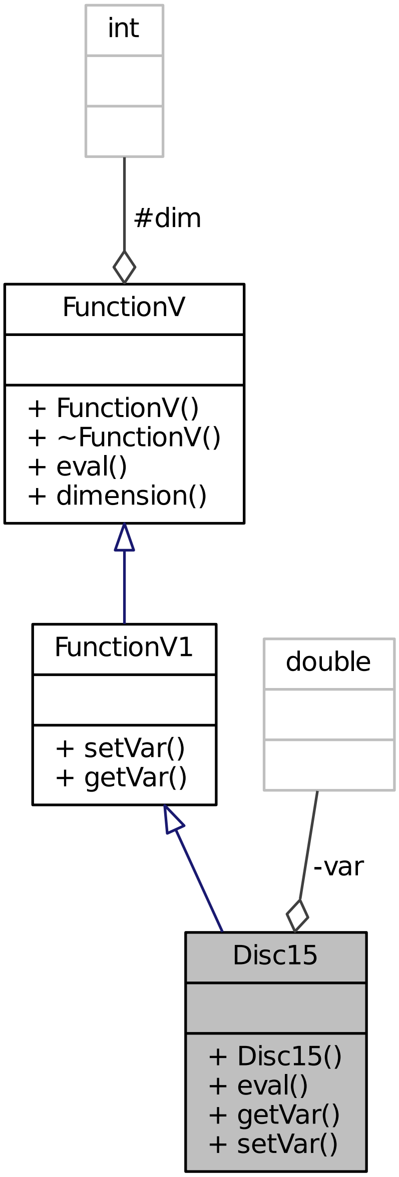
7.11 Disc15 Class Reference

```
#include <Disc15.hpp>
```

Inheritance diagram for Disc15:



Collaboration diagram for Disc15:



Public Member Functions

- Disc15 (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.11.1 Constructor & Destructor Documentation

7.11.1.1 Disc15::Disc15 (Idouble *b*) [inline]

7.11.2 Member Function Documentation

7.11.2.1 Idouble Disc15::eval (int *d,* Idouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.11.2.2 Idouble Disc15::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.11.2.3 void Disc15::setVar (Idouble *b*) [inline], [virtual]

Implements FunctionV1.

7.11.3 Member Data Documentation

7.11.3.1 Idouble Disc15::var [private]

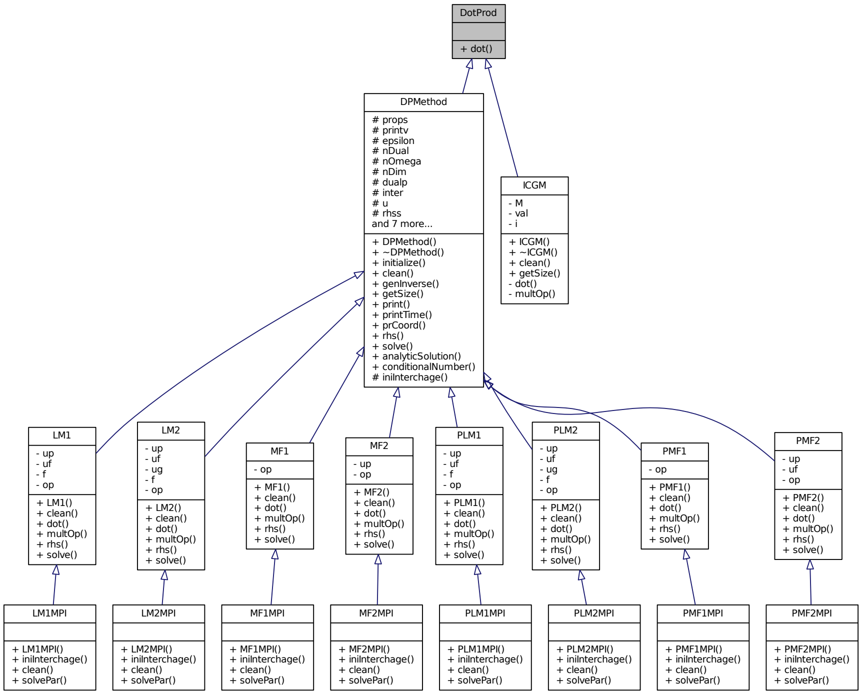
The documentation for this class was generated from the following file:

- Disc15.hpp

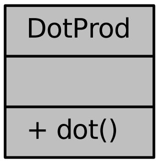
7.12 DotProd Class Reference

```
#include <DotProd.hpp>
```

Inheritance diagram for DotProd:



Collaboration diagram for DotProd:



Public Member Functions

- virtual ldouble dot (ldouble ∗x, ldouble ∗y)=0

7.12.1 Member Function Documentation

7.12.1.1 virtual ldouble DotProd::dot (ldouble ∗ *x*, ldouble ∗ *y*) [pure virtual]

Implemented in LM2, PLM1, PLM2, LM1, MF2, PMF2, PMF1, MF1, and ICGM.

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

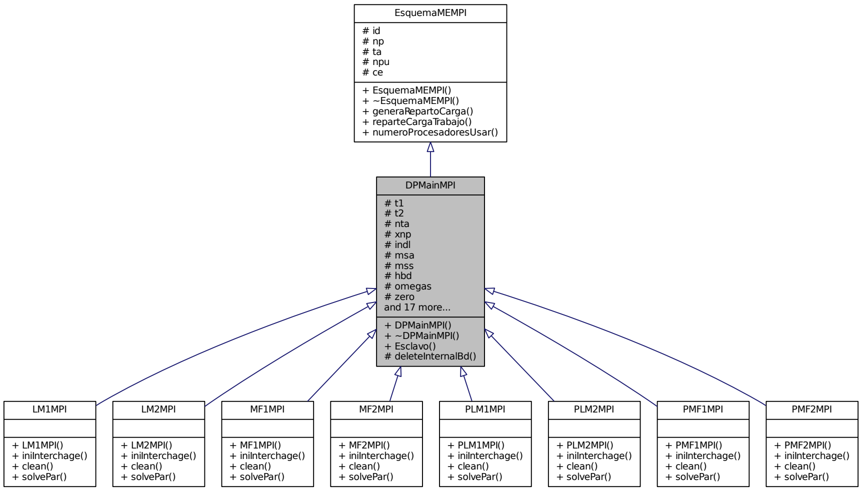
- DotProd.hpp

7.13 DPMainMPI Class Reference

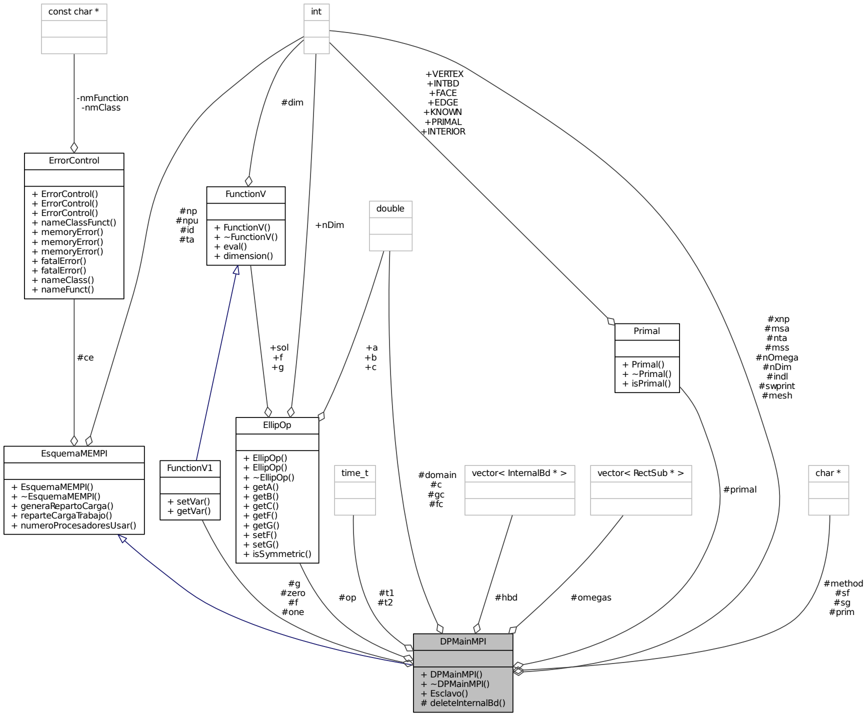
Clase base para definir a los metodos DVS-DDM.

`#include <DPMainMPI.hpp>`

Inheritance diagram for DPMainMPI:



Collaboration diagram for DPMainMPI:



Public Member Functions

- DPMainMPI (int id, int np, PropDef &props, EllipOp &op)

  Constructor de la clase.
- ∼DPMainMPI ()

  Destructor de la clase.
- void Esclavo (void)

  Esclavo.

Protected Member Functions

- void deleteInternalBd (void)

Protected Attributes

- time_t t1

  Tiempo inicial.
- time_t t2

  Tiempo final.
- int nta

  Nmero de tareas por nodo esclavo.

- int xnp

  Nmero de esclavo en el que estara la tarea.
- int indl

  Nmero de tarea dentro del esclavo.
- int msa [10]

  Arreglo para recibir mensajes.
- int mss [10]

  Arreglo para enviar mensajes.
- vector$<$ InternalBd $*$ $>$ hbd
- vector$<$ RectSub $*$ $>$ omegas
- FunctionV1 $*$ zero
- FunctionV1 $*$ one
- FunctionV1 $*$ f
- FunctionV1 $*$ g
- char $*$ sf
- char $*$ sg
- ldouble fc
- ldouble gc
- int $*$ mesh
- char $*$ prim
- char $*$ method
- int swprint
- ldouble $**$ domain
- ldouble c
- Primal $*$ primal
- EllipOp $*$ op
- int nDim
- int nOmega

7.13.1 Detailed Description

Clase base para definir a los metodos DVS-DDM.

Clase base para definir a los metodos DVS-DDM en paralelo en donde se definen las operaciones que realizaran los nodos esclavos del esquema Mestro-Esclavo y la inicializacion de la parte paralela de la ejecucion

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.13.2 Constructor & Destructor Documentation

7.13.2.1 DPMainMPI::DPMainMPI (int *id*, int *np*, PropDef & *props*, EllipOp & *op*)

Constructor de la clase.

7.13.2.2 DPMainMPI::∼DPMainMPI ( )

Destructor de la clase.

7.13.3 Member Function Documentation

7.13.3.1 void DPMainMPI::deleteInternalBd (void) [protected]

7.13.3.2 void DPMainMPI::Esclavo (void)

Esclavo.

7.13.4 Member Data Documentation

7.13.4.1 ldouble DPMainMPI::c [protected]

7.13.4.2 ldouble∗∗ DPMainMPI::domain [protected]

7.13.4.3 FunctionV1∗ DPMainMPI::f [protected]

7.13.4.4 ldouble DPMainMPI::fc [protected]

7.13.4.5 FunctionV1 ∗ DPMainMPI::g [protected]

7.13.4.6 ldouble DPMainMPI::gc [protected]

7.13.4.7 vector<InternalBd∗> DPMainMPI::hbd [protected]

7.13.4.8 int DPMainMPI::indl [protected]

Nmero de tarea dentro del esclavo.

7.13.4.9 int∗ DPMainMPI::mesh [protected]

7.13.4.10 char∗ DPMainMPI::method [protected]

7.13.4.11 int DPMainMPI::msa[10] [protected]

Arreglo para recibir mensajes.

7.13.4.12 int DPMainMPI::mss[10] [protected]

Arreglo para enviar mensajes.

7.13.4.13 int DPMainMPI::nDim [protected]

7.13.4.14 int DPMainMPI::nOmega [protected]

7.13.4.15 int DPMainMPI::nta [protected]

Nmero de tareas por nodo esclavo.

7.13.4.16 vector<RectSub*> DPMainMPI::omegas [protected]

7.13.4.17 FunctionV1 * DPMainMPI::one [protected]

7.13.4.18 EllipOp* DPMainMPI::op [protected]

7.13.4.19 char* DPMainMPI::prim [protected]

7.13.4.20 Primal* DPMainMPI::primal [protected]

7.13.4.21 char* DPMainMPI::sf [protected]

7.13.4.22 char * DPMainMPI::sg [protected]

7.13.4.23 int DPMainMPI::swprint [protected]

7.13.4.24 time_t DPMainMPI::t1 [protected]

Tiempo inicial.

7.13.4.25 time_t DPMainMPI::t2 [protected]

Tiempo final.

7.13.4.26 int DPMainMPI::xnp [protected]

Nmero de esclavo en el que estara la tarea.

7.13.4.27 FunctionV1* DPMainMPI::zero [protected]

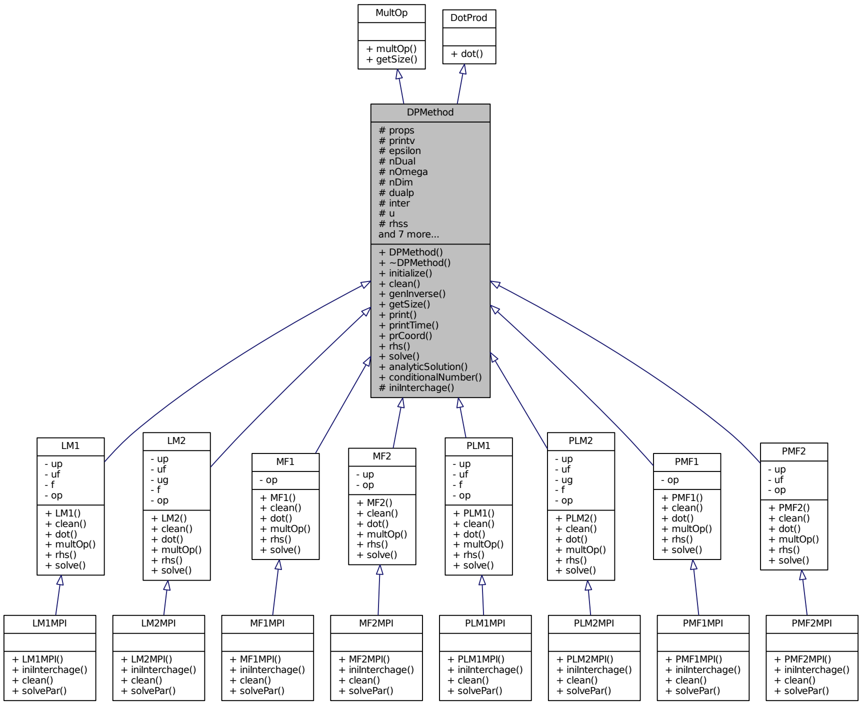
The documentation for this class was generated from the following files:

- DPMainMPI.hpp
- DPMainMPI.cpp

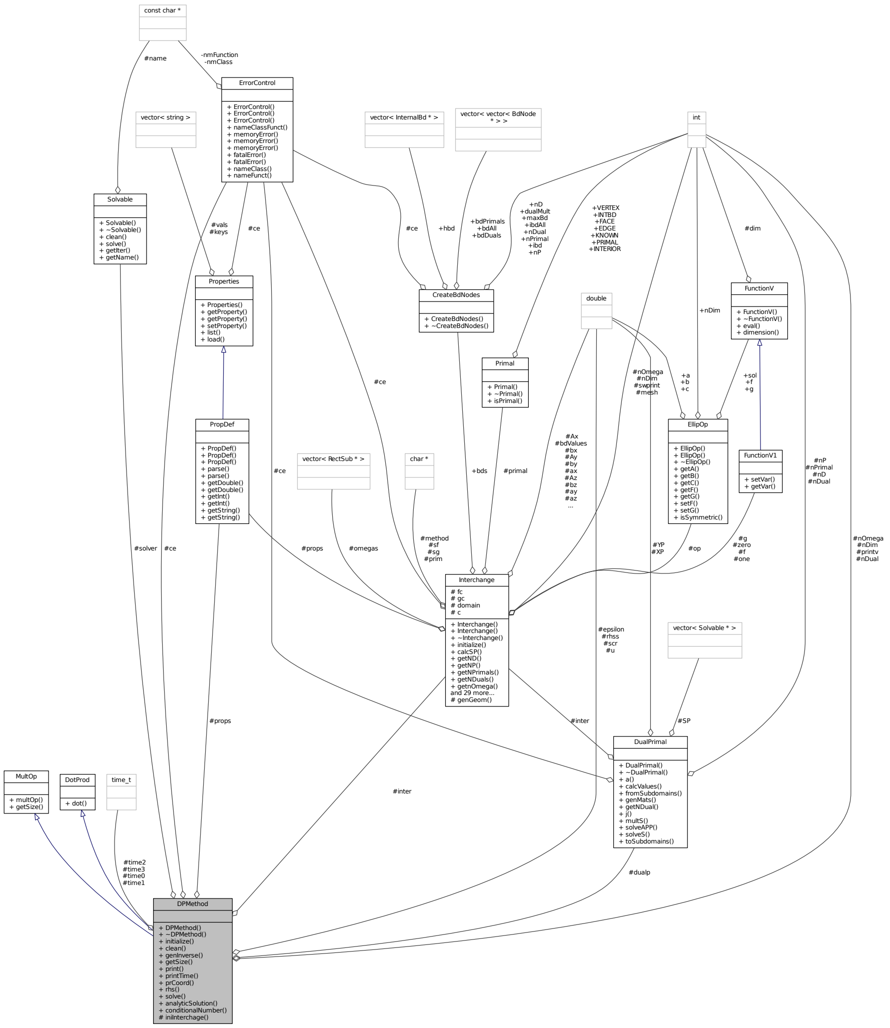
7.14 DPMethod Class Reference

```
#include <DPMethod.hpp>
```

Inheritance diagram for DPMethod:



Collaboration diagram for DPMethod:



Public Member Functions

- DPMethod (PropDef &props)
- virtual ∼DPMethod ()
- void initialize (void)
- virtual void clean (void)=0
- void genInverse (int type)
- int getSize (void)

  vector size
- void print (ldouble ∗u)
- void printTime (void)
- const char ∗ prCoord (ldouble ∗x)
- virtual void rhs (void)=0
- virtual void solve (void)=0
- double analyticSolution (double ∗x)

- void conditionalNumber (bool symetric)

  Calcula el numero de condicionamiento.

Protected Member Functions

- virtual void iniInterchage (void)

  Inicializa los subdominios.

Protected Attributes

- PropDef ∗ props
- int printv
- ldouble epsilon
- int nDual
- int nOmega
- int nDim
- DualPrimal ∗ dualp
- Interchange ∗ inter
- ldouble ∗ u
- ldouble ∗ rhss
- ldouble ∗ scr
- Solvable ∗ solver
- time_t time0
- time_t time1
- time_t time2
- time_t time3
- ErrorControl ce

  Control de errores.

7.14.1 Constructor & Destructor Documentation

7.14.1.1 DPMethod::DPMethod (PropDef & *props*) [inline]

7.14.1.2 virtual DPMethod::∼DPMethod ( ) [inline], [virtual]

7.14.2 Member Function Documentation

7.14.2.1 double DPMethod::analyticSolution (double ∗ *x*)

7.14.2.2 virtual void DPMethod::clean (void) [pure virtual]

Implemented in LM2MPI, MF1MPI, MF2MPI, PLM1MPI, PLM2MPI, PMF2MPI, LM1MPI, PMF1MPI, LM2, PLM1, PLM2, LM1, MF2, PMF2, PMF1, and MF1.

7.14.2.3 void DPMethod::conditionalNumber (bool *symetric*)

Calcula el numero de condicionamiento.

7.14.2.4 void DPMethod::genInverse (int *type*)

7.14.2.5 int DPMethod::getSize (void) [inline], [virtual]

vector size

Implements MultOp.

7.14.2.6 virtual void DPMethod::iniInterchage (void) [inline], [protected], [virtual]

Inicializa los subdominios.

Reimplemented in LM2MPI, MF1MPI, MF2MPI, PLM1MPI, PLM2MPI, PMF2MPI, LM1MPI, and PMF1MPI.

7.14.2.7 void DPMethod::initialize (void)

7.14.2.8 const char ∗ DPMethod::prCoord (ldouble ∗ *x*)

7.14.2.9 void DPMethod::print (ldouble ∗ *u*)

7.14.2.10 void DPMethod::printTime (void)

7.14.2.11 virtual void DPMethod::rhs (void) [pure virtual]

Implemented in LM2, PLM1, PLM2, LM1, MF2, PMF2, PMF1, and MF1.

7.14.2.12 virtual void DPMethod::solve (void) [pure virtual]

Implemented in LM2, PLM1, PLM2, LM1, MF2, PMF2, PMF1, and MF1.

7.14.3 Member Data Documentation

7.14.3.1 ErrorControl DPMethod::ce [protected]

Control de errores.

7.14.3.2 DualPrimal∗ DPMethod::dualp [protected]

7.14.3.3 ldouble DPMethod::epsilon [protected]

7.14.3.4 Interchange∗ DPMethod::inter [protected]

7.14.3.5 int DPMethod::nDim [protected]

7.14.3.6 int DPMethod::nDual [protected]

7.14.3.7 int DPMethod::nOmega [protected]

7.14.3.8 int DPMethod::printv [protected]

7.14.3.9 PropDef∗ DPMethod::props [protected]

7.14.3.10 Idouble∗ DPMethod::rhss [protected]

7.14.3.11 Idouble∗ DPMethod::scr [protected]

7.14.3.12 Solvable∗ DPMethod::solver [protected]

7.14.3.13 time_t DPMethod::time0 [protected]

7.14.3.14 time_t DPMethod::time1 [protected]

7.14.3.15 time_t DPMethod::time2 [protected]

7.14.3.16 time_t DPMethod::time3 [protected]

7.14.3.17 Idouble∗ DPMethod::u [protected]

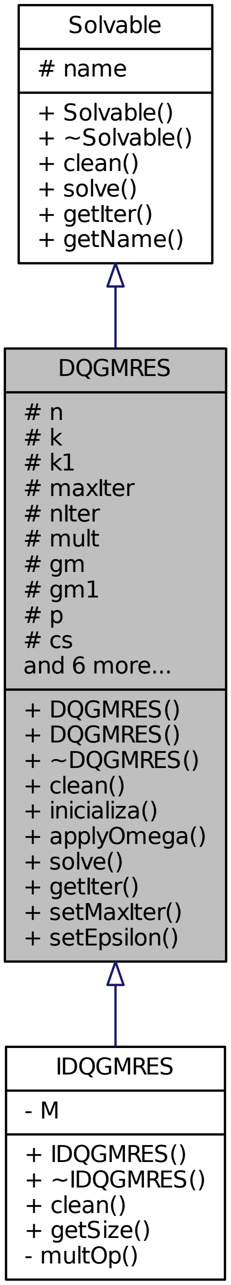
The documentation for this class was generated from the following files:

- DPMethod.hpp
- DPMethod.cpp

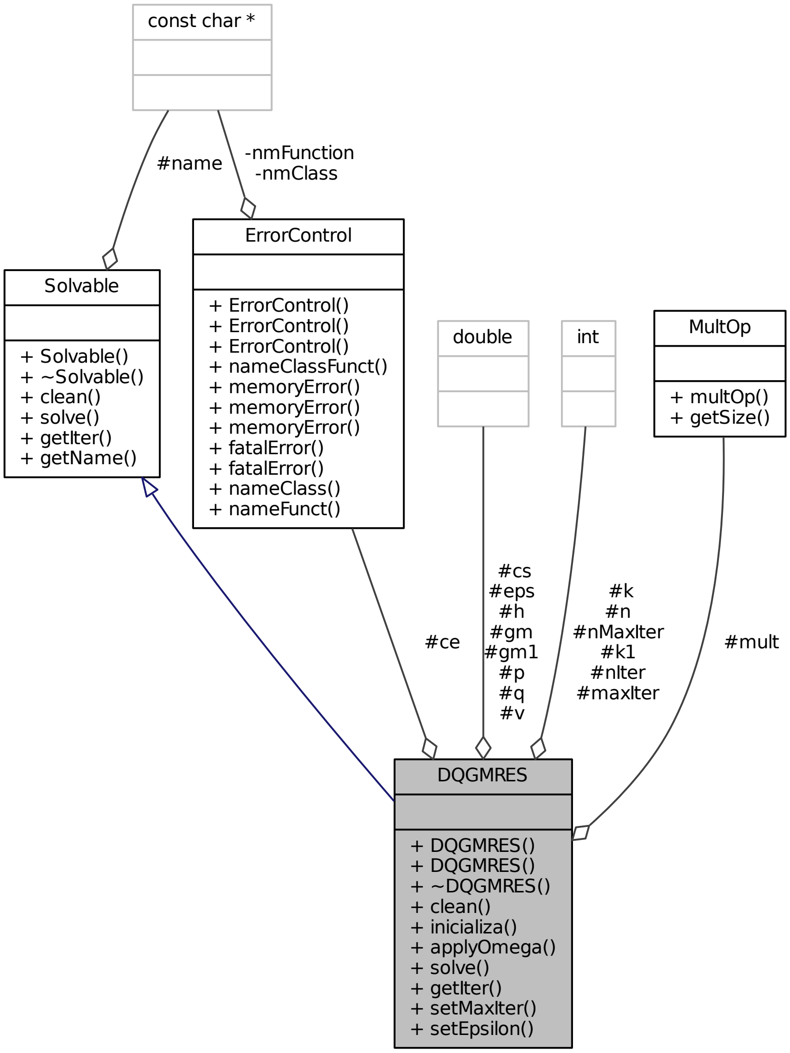
7.15 DQGMRES Class Reference

```
#include <DQGMRES.hpp>
```

Inheritance diagram for DQGMRES:



Collaboration diagram for DQGMRES:



Public Member Functions

- DQGMRES (MultOp &mult, int k, ldouble eps)
- DQGMRES (void)
- ∼DQGMRES (void)
- void clean (void)
- void inicializa (void)
- void applyOmega (int m)
- void solve (ldouble ∗x, ldouble ∗b)
- int getIter (void)
- void setMaxIter (int nmi)
- void setEpsilon (ldouble ep)

Protected Attributes

- int n
- int k
- int k1
- int maxIter
- int nIter
- MultOp ∗ mult
- ldouble gm
- ldouble gm1
- ldouble ∗∗ p
- ldouble ∗∗ cs
- ldouble ∗∗ h
- ldouble ∗∗ q
- ldouble ∗ v
- ldouble eps
- int nMaxIter
- ErrorControl ce

7.15.1 Constructor & Destructor Documentation

7.15.1.1 DQGMRES::DQGMRES (MultOp & *mult*, int *k*, ldouble *eps*) `[inline]`

7.15.1.2 DQGMRES::DQGMRES (void) `[inline]`

7.15.1.3 DQGMRES::∼DQGMRES (void) `[inline]`

7.15.2 Member Function Documentation

7.15.2.1 void DQGMRES::applyOmega (int *m*)

7.15.2.2 void DQGMRES::clean (void) `[inline]`, `[virtual]`

Implements Solvable.

Reimplemented in IDQGMRES.

7.15.2.3 int DQGMRES::getIter (void) `[inline]`, `[virtual]`

Implements Solvable.

7.15.2.4 void DQGMRES::inicializa (void)

7.15.2.5 void DQGMRES::setEpsilon (ldouble *ep*) `[inline]`

7.15.2.6 void DQGMRES::setMaxIter (int *nmi*) `[inline]`

7.15.2.7 void DQGMRES::solve (ldouble ∗ *x*, ldouble ∗ *b*) `[virtual]`

Implements Solvable.

7.15.3 Member Data Documentation

7.15.3.1 ErrorControl DQGMRES::ce [protected]

7.15.3.2 ldouble** DQGMRES::cs [protected]

7.15.3.3 ldouble DQGMRES::eps [protected]

7.15.3.4 ldouble DQGMRES::gm [protected]

7.15.3.5 ldouble DQGMRES::gm1 [protected]

7.15.3.6 ldouble** DQGMRES::h [protected]

7.15.3.7 int DQGMRES::k [protected]

7.15.3.8 int DQGMRES::k1 [protected]

7.15.3.9 int DQGMRES::maxIter [protected]

7.15.3.10 MultOp* DQGMRES::mult [protected]

7.15.3.11 int DQGMRES::n [protected]

7.15.3.12 int DQGMRES::nIter [protected]

7.15.3.13 int DQGMRES::nMaxIter [protected]

7.15.3.14 ldouble** DQGMRES::p [protected]

7.15.3.15 ldouble** DQGMRES::q [protected]

7.15.3.16 ldouble* DQGMRES::v [protected]

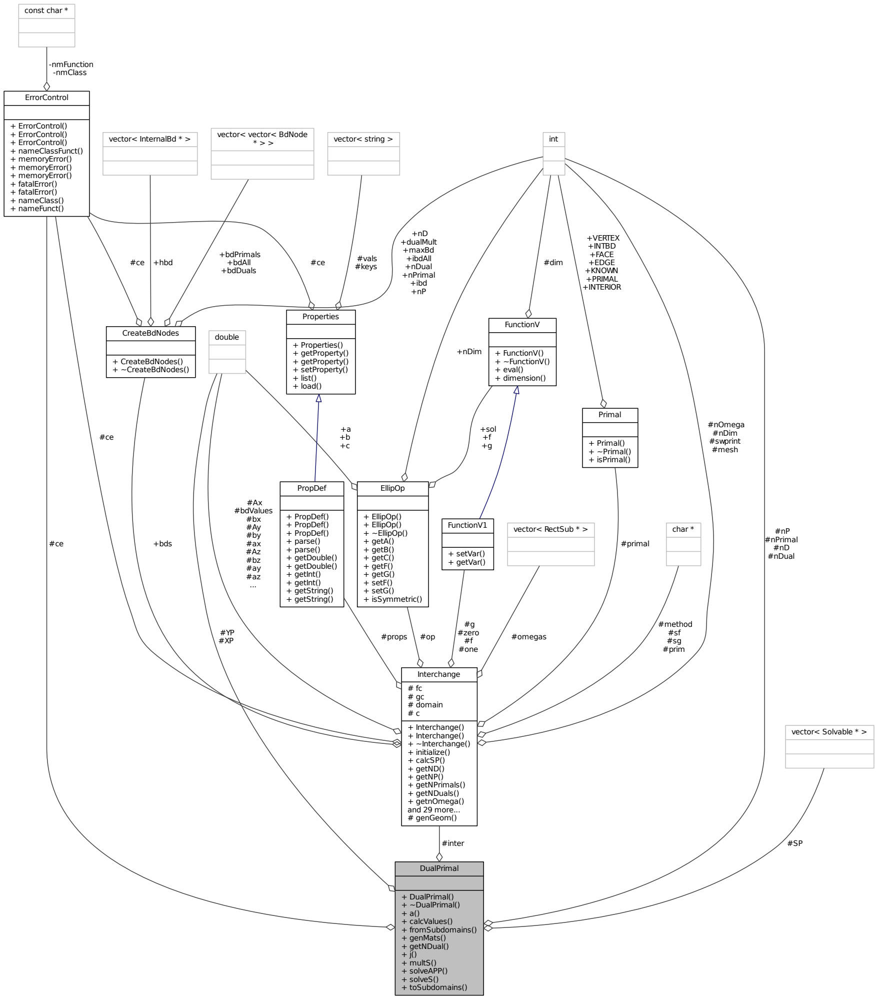
The documentation for this class was generated from the following files:

- DQGMRES.hpp
- DQGMRES.cpp

7.16 DualPrimal Class Reference

```
#include <DualPrimal.hpp>
```

Collaboration diagram for DualPrimal:



Public Member Functions

- DualPrimal (Interchange &inter)
- ∼DualPrimal (void)
- void a (ldouble ∗u, ldouble ∗v)
- void calcValues (ldouble ∗u)
- void fromSubdomains (int sc, ldouble ∗u)
- void genMats (void)
- int getNDual (void)
- void j (ldouble ∗u, ldouble ∗v)
- void multS (ldouble ∗u, ldouble ∗v)
- void solveAPP (int sp, int sc1, int sc2, int sc3)
- void solveS (ldouble ∗u, ldouble ∗v)
- void toSubdomains (int sc, ldouble ∗u)

Protected Attributes

- int nPrimal
- int nDual
- Interchange ∗ inter
- vector< Solvable ∗ > SP
- ldouble ∗ XP
- ldouble ∗ YP
- int nD
- int nP
- ErrorControl ce

  Control de errores.

7.16.1 Constructor & Destructor Documentation

7.16.1.1 DualPrimal::DualPrimal (**Interchange & *inter***)

7.16.1.2 DualPrimal::∼DualPrimal (void) `[inline]`

7.16.2 Member Function Documentation

7.16.2.1 void DualPrimal::a (**ldouble ∗ *u*, ldouble ∗ *v***)

7.16.2.2 void DualPrimal::calcValues (**ldouble ∗ *u***)

7.16.2.3 void DualPrimal::fromSubdomains (int *sc*, **ldouble ∗ *u***)

7.16.2.4 void DualPrimal::genMats (void)

7.16.2.5 int DualPrimal::getNDual (void) `[inline]`

7.16.2.6 void DualPrimal::j (**ldouble ∗ *u*, ldouble ∗ *v***)

7.16.2.7 void DualPrimal::multS (**ldouble ∗ *u*, ldouble ∗ *v***)

7.16.2.8 void DualPrimal::solveAPP (int *sp*, int *sc1*, int *sc2*, int *sc3*)

7.16.2.9 void DualPrimal::solveS (**ldouble ∗ *u*, ldouble ∗ *v***)

7.16.2.10 void DualPrimal::toSubdomains (int *sc*, **ldouble ∗ *u***)

7.16.3 Member Data Documentation

7.16.3.1 ErrorControl DualPrimal::ce `[protected]`

Control de errores.

7.16.3.2 Interchange∗ DualPrimal::inter `[protected]`

7.16.3.3 int DualPrimal::nD `[protected]`

7.16.3.4 int DualPrimal::nDual [protected]

7.16.3.5 int DualPrimal::nP [protected]

7.16.3.6 int DualPrimal::nPrimal [protected]

7.16.3.7 vector<Solvable*> DualPrimal::SP [protected]

7.16.3.8 Idouble* DualPrimal::XP [protected]

7.16.3.9 Idouble* DualPrimal::YP [protected]

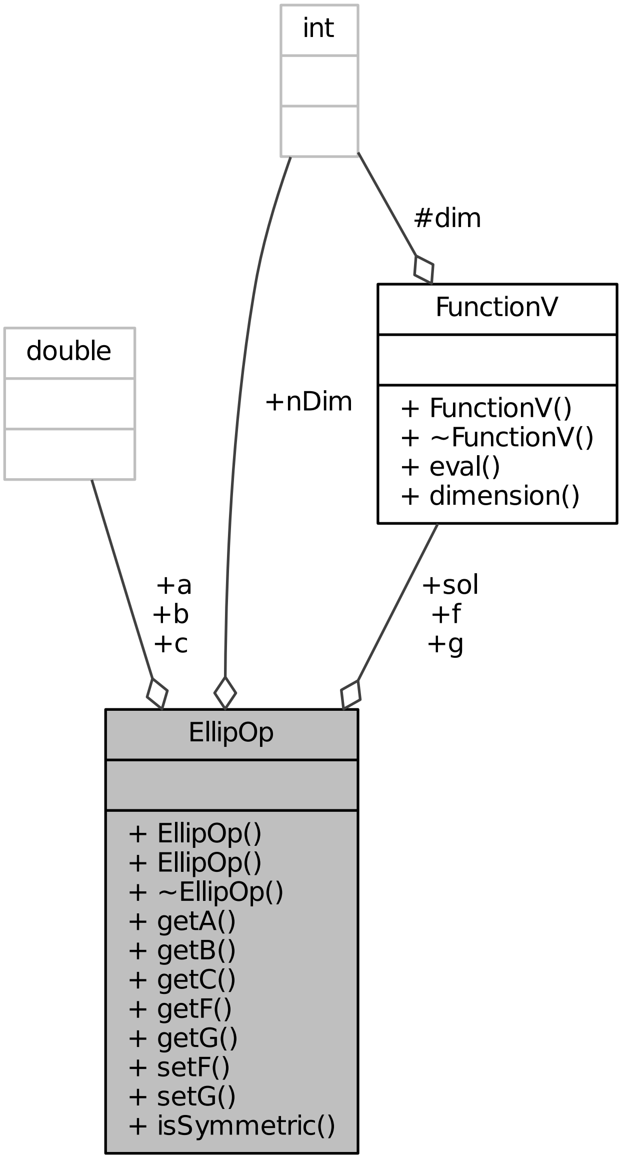
The documentation for this class was generated from the following files:

- DualPrimal.hpp
- DualPrimal.cpp

7.17 EllipOp Class Reference

```
#include <EllipOp.hpp>
```

Collaboration diagram for EllipOp:



Public Member Functions

- EllipOp (int nDim, ldouble ∗a, ldouble ∗b, ldouble c, FunctionV &f, FunctionV &g, FunctionV &sol)
- EllipOp (int nDim, ldouble ∗a, ldouble ∗b, ldouble c)
- ∼EllipOp ()
- ldouble ∗ getA (void)

- ldouble ∗ getB (void)
- ldouble getC (void)
- FunctionV ∗ getF (void)
- FunctionV ∗ getG (void)
- void setF (FunctionV &f)
- void setG (FunctionV &g)
- bool isSymmetric (void)

Public Attributes

- int nDim
- ldouble ∗ a
- ldouble ∗ b
- ldouble c
- FunctionV ∗ f
- FunctionV ∗ g
- FunctionV ∗ sol

7.17.1 Constructor & Destructor Documentation

7.17.1.1 EllipOp::EllipOp (int *nDim*, ldouble ∗ *a*, ldouble ∗ *b*, ldouble *c*, FunctionV & *f*, FunctionV & *g*, FunctionV & *sol*) `[inline]`

7.17.1.2 EllipOp::EllipOp (int *nDim*, ldouble ∗ *a*, ldouble ∗ *b*, ldouble *c*) `[inline]`

7.17.1.3 EllipOp::∼EllipOp ( ) `[inline]`

7.17.2 Member Function Documentation

7.17.2.1 ldouble∗ EllipOp::getA (void) `[inline]`

7.17.2.2 ldouble∗ EllipOp::getB (void) `[inline]`

7.17.2.3 ldouble EllipOp::getC (void) `[inline]`

7.17.2.4 FunctionV∗ EllipOp::getF (void) `[inline]`

7.17.2.5 FunctionV∗ EllipOp::getG (void) `[inline]`

7.17.2.6 bool EllipOp::isSymmetric (void) `[inline]`

7.17.2.7 void EllipOp::setF (FunctionV & *f*) `[inline]`

7.17.2.8 void EllipOp::setG (FunctionV & *g*) `[inline]`

7.17.3 Member Data Documentation

7.17.3.1 ldouble∗ EllipOp::a

7.17.3.2 ldouble∗ EllipOp::b

7.17.3.3 Idouble EllipOp::c

7.17.3.4 FunctionV∗ EllipOp::f

7.17.3.5 FunctionV∗ EllipOp::g

7.17.3.6 int EllipOp::nDim

7.17.3.7 FunctionV∗ EllipOp::sol

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

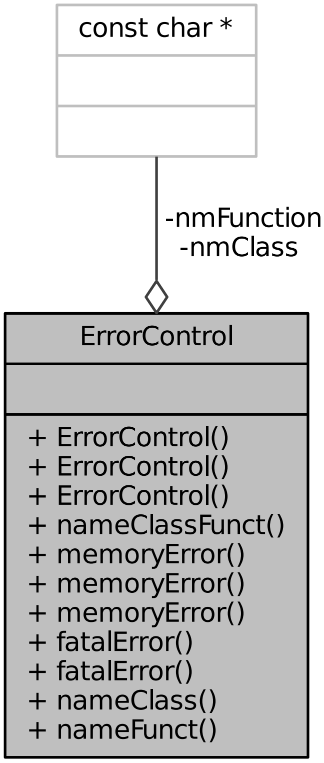
- EllipOp.hpp

7.18 ErrorControl Class Reference

Error Control.

```
#include <ErrorControl.hpp>
```

Collaboration diagram for ErrorControl:



Public Member Functions

- ErrorControl (void)
- ErrorControl (const char ∗clas)
- ErrorControl (const char ∗clas, const char ∗fun)
- void nameClassFunct (const char ∗clas, const char ∗func)
- void memoryError (const char ∗var)
- void memoryError (const char ∗var, int i)
- void memoryError (const char ∗var, const char ∗func)
- void fatalError (int cod)
- void fatalError (int cod, const char ∗txt)
- void nameClass (const char ∗clas)
- void nameFunct (const char ∗func)

Private Attributes

- const char ∗ nmClass

  Name of class.
- const char ∗ nmFunction

  Name of function.

7.18.1 Detailed Description

Error Control.

Author

Antonio Carrillo

Date

Winter 2010

Version

0.0.1

Bug No errors detected

Todo Exception handling

7.18.2 Constructor & Destructor Documentation

7.18.2.1 ErrorControl::ErrorControl (void)

Class Constructor

7.18.2.2 ErrorControl::ErrorControl (const char ∗ *clas*)

Class Constructor

Parameters

clas	Class name

7.18.2.3 ErrorControl::ErrorControl (const char ∗ *clas,* const char ∗ *fun*)

Class Constructor

Parameters

clas	Class name
fun	Function name

7.18.3 Member Function Documentation

7.18.3.1 void ErrorControl::fatalError (int *cod*)

Fatal error.

Parameters

cod	Error code

7.18.3.2 void ErrorControl::fatalError (int *cod,* const char ∗ *txt*)

Fatal error.

Parameters

cod	Error code
txt	Text for user

7.18.3.3 void ErrorControl::memoryError (const char ∗ *var*)

No memory for this request

Parameters

var	Var name

7.18.3.4 void ErrorControl::memoryError (const char ∗ *var,* int *i*)

No memory for this request

Parameters

var	Var name
i	Index number

7.18.3.5 void ErrorControl::memoryError (const char * *var,* const char * *func*)

No memory for this request

Parameters

var	Var name
func	Function name

7.18.3.6 void ErrorControl::nameClass (const char ∗ *clas*)

Set name of class

Parameters

clas	Class name

7.18.3.7 void ErrorControl::nameClassFunct (const char ∗ *clas,* const char ∗ *func*)

Name of class and function

Parameters

clas	Class name
func	Function name

7.18.3.8 void ErrorControl::nameFunct (const char ∗ *func*)

Set name of function

Parameters

func	Function name

7.18.4 Member Data Documentation

7.18.4.1 const char∗ ErrorControl::nmClass `[private]`

Name of class.

7.18.4.2 const char∗ ErrorControl::nmFunction `[private]`

Name of function.

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

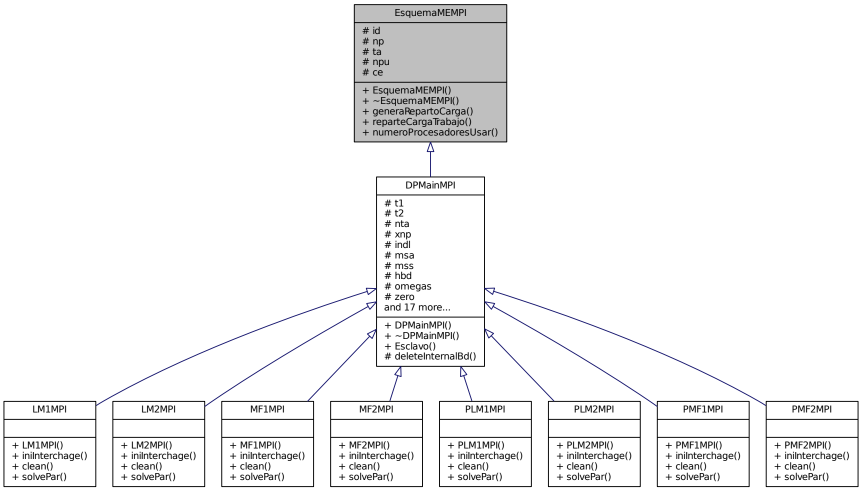
- ErrorControl.hpp
- ErrorControl.cpp

7.19 EsquemaMEMPI Class Reference

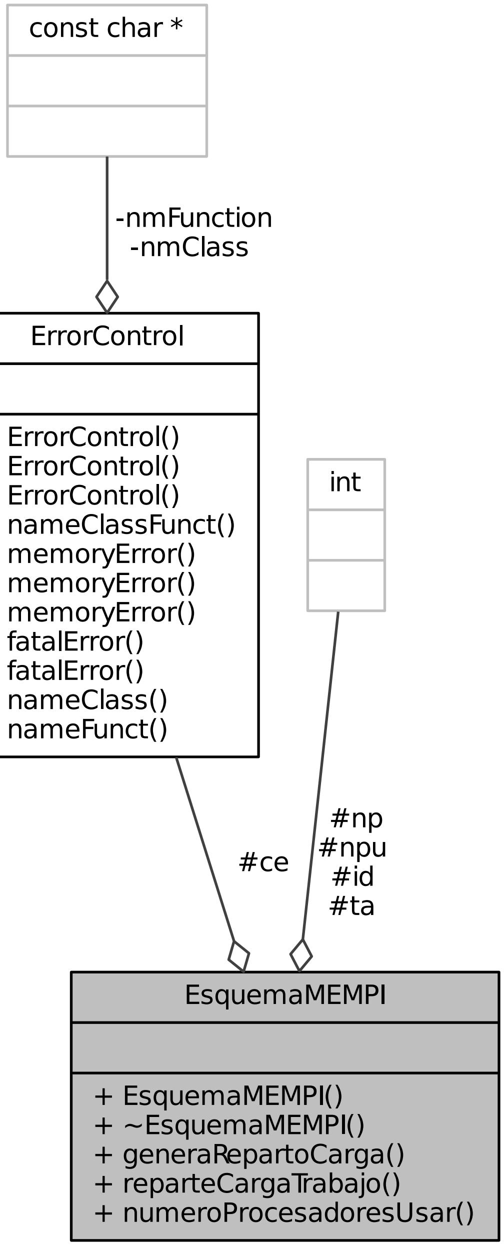
Clase base para definir el Esquema Maestro-Esclavo en MPI.

`#include <EsquemaMEMPI.hpp>`

Inheritance diagram for EsquemaMEMPI:



Collaboration diagram for EsquemaMEMPI:



Public Member Functions

- EsquemaMEMPI (int id, int np)

    Constructor de la clase.

- ∼EsquemaMEMPI ()

    Destructor de la clase.
- void generaRepartoCarga (int n)

    Genera el reparto de carga.
- void reparteCargaTrabajo (int &np, int &ind, int tarea)

    Reparte la carga de trabajo entre los nodos esclavos.
- int numeroProcesadoresUsar (void)

    Retorna el numero de procesadores a usar por el esquema M-E.

Protected Attributes

- int id

    Identificador.
- int np

    Numero de procesadores.
- int ∗ ta

    Numero de tareas por nodo esclavo.
- int npu

    Numero de nodos esclavos a utilizar (los que tienen carga)
- ErrorControl ce

    Control de errores.

7.19.1 Detailed Description

Clase base para definir el Esquema Maestro-Esclavo en MPI.

Clase base para definir el Esquema Maestro-Esclavo para programar en paralelo mediante el paso de mensajes usando MPI, donde el primer procesador (id = 0) es el nodo mestro y el resto son los nodos esclavos. Las tareas se pueden repartir de manara que subdominios contiguos queden en un mismo nodo esclavo o queden en distinto nodo esclavo.

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.19.2 Constructor & Destructor Documentation

7.19.2.1 EsquemaMEMPI::EsquemaMEMPI (int *id*, int *np*) [inline]

Constructor de la clase.

Parameters

id	Identificador
np	Numero de procesadores

7.19.2.2 EsquemaMEMPI::∼EsquemaMEMPI ( ) [inline]

Destructor de la clase.

7.19.3 Member Function Documentation

7.19.3.1 void EsquemaMEMPI::generaRepartoCarga (int *n*)

Genera el reparto de carga.

Parameters

n	Numero de trabajos

7.19.3.2 int EsquemaMEMPI::numeroProcesadoresUsar (void) [inline]

Retorna el numero de procesadores a usar por el esquema M-E.

Returns

> Numero de procesadores a usar dentro del esquema Maestro-Esclavo

7.19.3.3 void EsquemaMEMPI::reparteCargaTrabajo (int & *np,* int & *ind,* int *tarea*)

Reparte la carga de trabajo entre los nodos esclavos.

Parameters

np	Numero de procesador esclavo
st	Indice de tarea dentro del nodo esclavo
tarea	Tarea la cual debe ser repartida

7.19.4 Member Data Documentation

7.19.4.1 ErrorControl EsquemaMEMPI::ce [protected]

Control de errores.

7.19.4.2 int EsquemaMEMPI::id [protected]

Identificador.

7.19.4.3 int EsquemaMEMPI::np [protected]

Numero de procesadores.

7.19.4.4 int EsquemaMEMPI::npu [protected]

Numero de nodos esclavos a utilizar (los que tienen carga)

7.19.4.5 int∗ EsquemaMEMPI::ta [protected]

Numero de tareas por nodo esclavo.

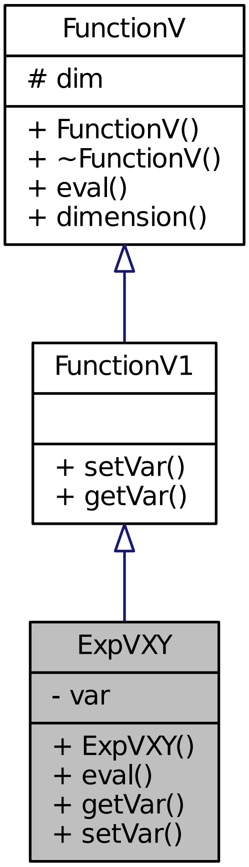
The documentation for this class was generated from the following files:

- EsquemaMEMPI.hpp
- EsquemaMEMPI.cpp

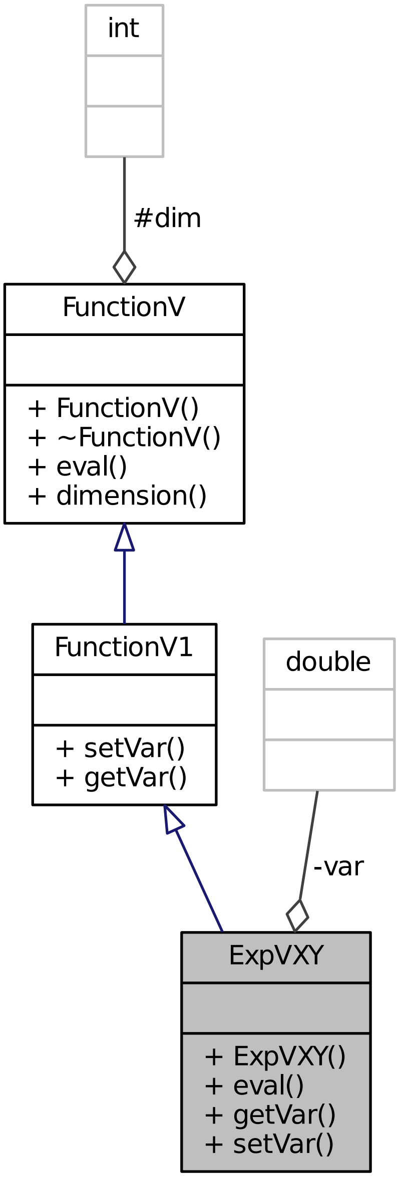
7.20 ExpVXY Class Reference

```
#include <ExpVXY.hpp>
```

Inheritance diagram for ExpVXY:



Collaboration diagram for ExpVXY:



Public Member Functions

- ExpVXY (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.20.1 Constructor & Destructor Documentation

7.20.1.1 ExpVXY::ExpVXY (Idouble *b*) `[inline]`

7.20.2 Member Function Documentation

7.20.2.1 Idouble ExpVXY::eval (int *d*, Idouble ∗ *x*) `[inline], [virtual]`

Implements FunctionV.

7.20.2.2 Idouble ExpVXY::getVar (void) `[inline], [virtual]`

Implements FunctionV1.

7.20.2.3 void ExpVXY::setVar (Idouble *b*) `[inline], [virtual]`

Implements FunctionV1.

7.20.3 Member Data Documentation

7.20.3.1 Idouble ExpVXY::var `[private]`

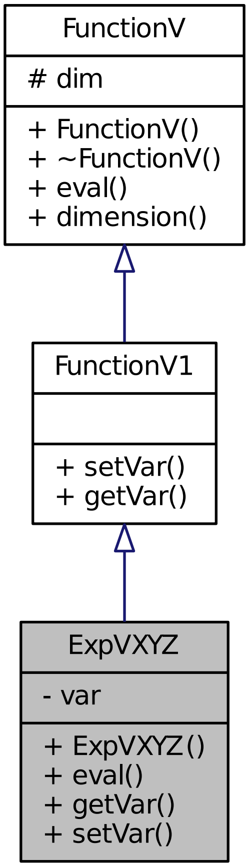
The documentation for this class was generated from the following file:

- ExpVXY.hpp

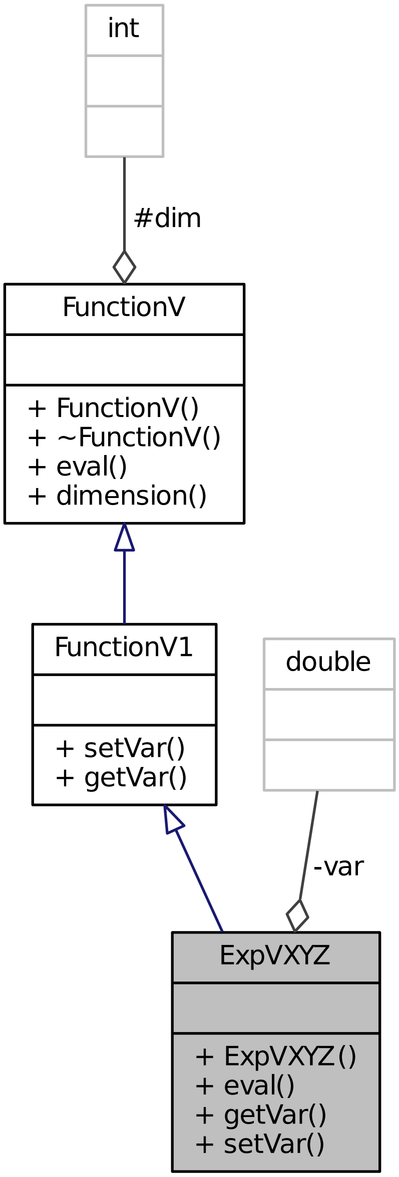
7.21 ExpVXYZ Class Reference

```
#include <ExpVXYZ.hpp>
```

Inheritance diagram for ExpVXYZ:



Collaboration diagram for ExpVXYZ:



Public Member Functions

- ExpVXYZ (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.21.1 Constructor & Destructor Documentation

7.21.1.1 ExpVXYZ::ExpVXYZ (Idouble *b*) `[inline]`

7.21.2 Member Function Documentation

7.21.2.1 Idouble ExpVXYZ::eval (int *d*, Idouble ∗ *x*) `[inline]`, `[virtual]`

Implements FunctionV.

7.21.2.2 Idouble ExpVXYZ::getVar (void) `[inline]`, `[virtual]`

Implements FunctionV1.

7.21.2.3 void ExpVXYZ::setVar (Idouble *b*) `[inline]`, `[virtual]`

Implements FunctionV1.

7.21.3 Member Data Documentation

7.21.3.1 Idouble ExpVXYZ::var `[private]`

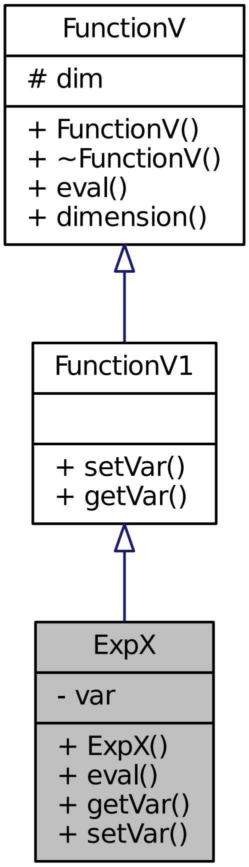
The documentation for this class was generated from the following file:

- ExpVXYZ.hpp

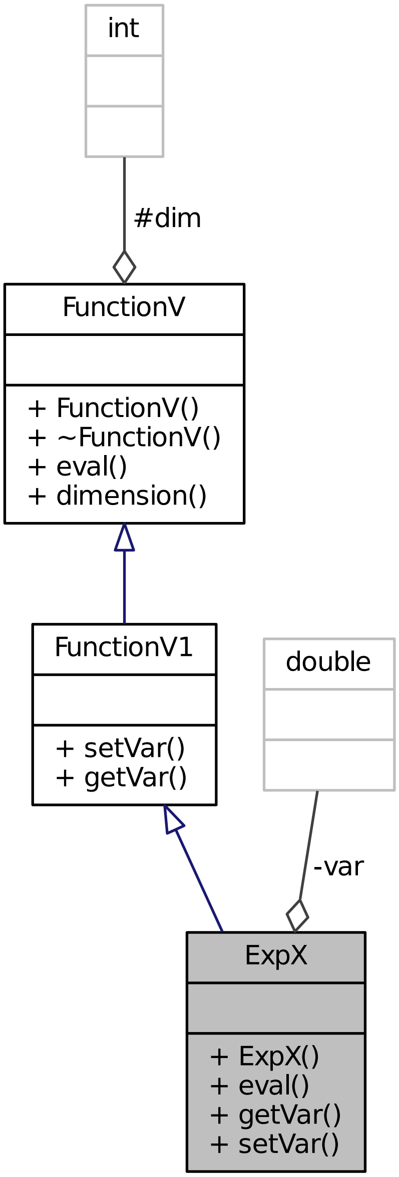
7.22 ExpX Class Reference

```
#include <ExpX.hpp>
```

Inheritance diagram for ExpX:



Collaboration diagram for ExpX:



Public Member Functions

- ExpX (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.22.1 Constructor & Destructor Documentation

7.22.1.1 ExpX::ExpX (Idouble *b*) [inline]

7.22.2 Member Function Documentation

7.22.2.1 Idouble ExpX::eval (int *d*, Idouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.22.2.2 Idouble ExpX::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.22.2.3 void ExpX::setVar (Idouble *b*) [inline], [virtual]

Implements FunctionV1.

7.22.3 Member Data Documentation

7.22.3.1 Idouble ExpX::var [private]

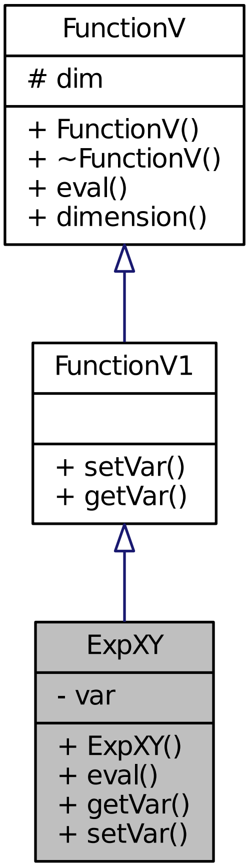
The documentation for this class was generated from the following file:

- ExpX.hpp

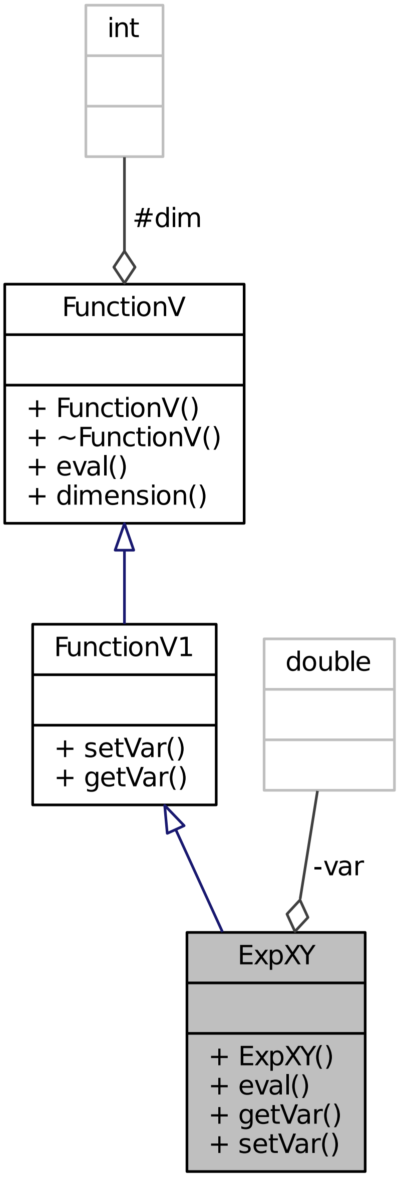
7.23 ExpXY Class Reference

```
#include <ExpXY.hpp>
```

Inheritance diagram for ExpXY:



Collaboration diagram for ExpXY:



Public Member Functions

- ExpXY (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- ldouble var

Additional Inherited Members

7.23.1 Constructor & Destructor Documentation

7.23.1.1 ExpXY::ExpXY (ldouble *b*) [inline]

7.23.2 Member Function Documentation

7.23.2.1 ldouble ExpXY::eval (int *d*, ldouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.23.2.2 ldouble ExpXY::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.23.2.3 void ExpXY::setVar (ldouble *b*) [inline], [virtual]

Implements FunctionV1.

7.23.3 Member Data Documentation

7.23.3.1 ldouble ExpXY::var [private]

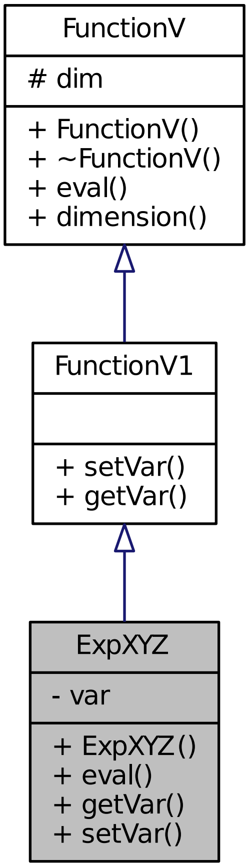
The documentation for this class was generated from the following file:

- ExpXY.hpp

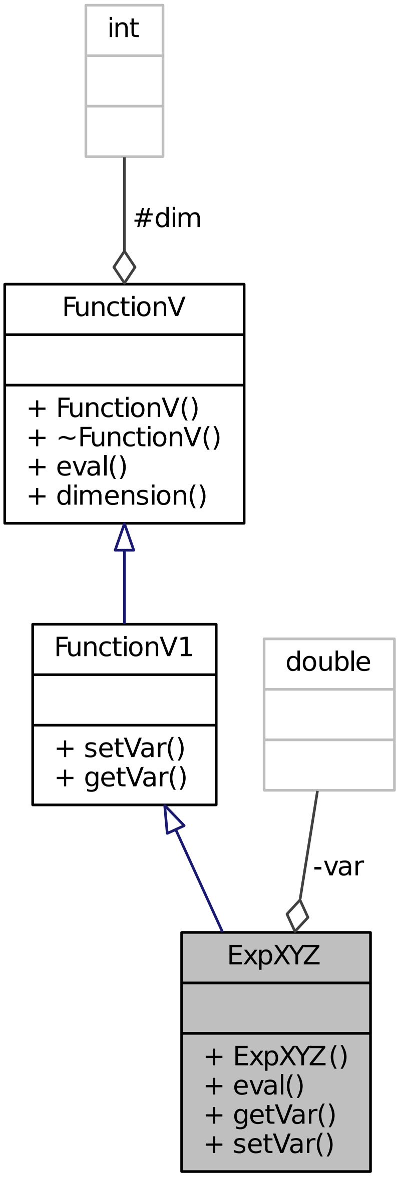
7.24 ExpXYZ Class Reference

```
#include <ExpXYZ.hpp>
```

Inheritance diagram for ExpXYZ:



Collaboration diagram for ExpXYZ:



Public Member Functions

- ExpXYZ (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- ldouble var

Additional Inherited Members

7.24.1 Constructor & Destructor Documentation

7.24.1.1 ExpXYZ::ExpXYZ (ldouble *b*) [inline]

7.24.2 Member Function Documentation

7.24.2.1 ldouble ExpXYZ::eval (int *d*, ldouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.24.2.2 ldouble ExpXYZ::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.24.2.3 void ExpXYZ::setVar (ldouble *b*) [inline], [virtual]

Implements FunctionV1.

7.24.3 Member Data Documentation

7.24.3.1 ldouble ExpXYZ::var [private]

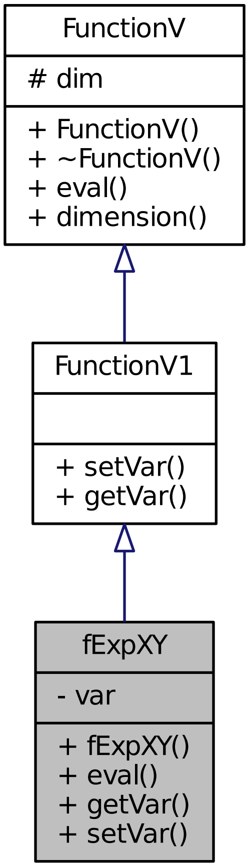
The documentation for this class was generated from the following file:

- ExpXYZ.hpp

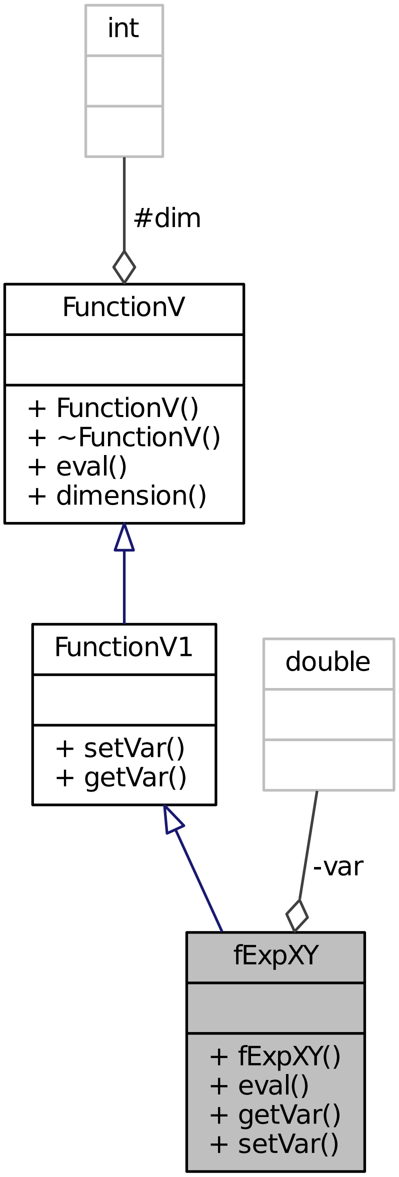
7.25 fExpXY Class Reference

```
#include <fExpXY.hpp>
```

Inheritance diagram for fExpXY:



Collaboration diagram for fExpXY:



Public Member Functions

- fExpXY (double b)
- double eval (int d, double ∗x)
- double getVar (void)
- void setVar (double b)

Private Attributes

- double var

Additional Inherited Members

7.25.1 Constructor & Destructor Documentation

7.25.1.1 fExpXY::fExpXY (double *b*) [inline]

7.25.2 Member Function Documentation

7.25.2.1 double fExpXY::eval (int *d*, double ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.25.2.2 double fExpXY::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.25.2.3 void fExpXY::setVar (double *b*) [inline], [virtual]

Implements FunctionV1.

7.25.3 Member Data Documentation

7.25.3.1 double fExpXY::var [private]

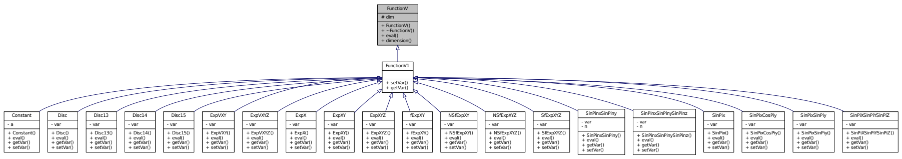
The documentation for this class was generated from the following file:

- fExpXY.hpp

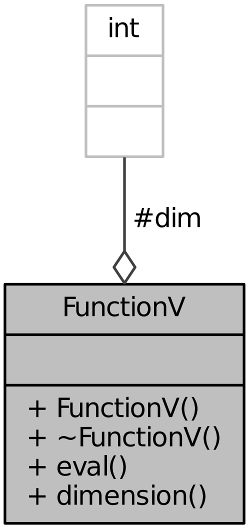
7.26 FunctionV Class Reference

```
#include <FunctionV.hpp>
```

Inheritance diagram for FunctionV:



Collaboration diagram for FunctionV:



Public Member Functions

- FunctionV (void)
- virtual ∼FunctionV ()
- virtual ldouble eval (int d, ldouble ∗x)=0
- void dimension (int d)

Protected Attributes

- int dim

7.26.1 Constructor & Destructor Documentation

7.26.1.1 FunctionV::FunctionV (void) [inline]

7.26.1.2 virtual FunctionV::∼FunctionV ( ) [inline], [virtual]

7.26.2 Member Function Documentation

7.26.2.1 void FunctionV::dimension (int *d*) [inline]

7.26.2.2 virtual Idouble FunctionV::eval (int *d*, Idouble ∗ *x*) [pure virtual]

Implemented in Disc13, Disc14, Disc15, NSfExpXYZ, SfExpXYZ, SinPinxSinPinySinPinz, SinPiXSinPiYSinPiZ, ExpXY, ExpXYZ, SinPinxSinPiny, SinPixCosPiy, SinPixSinPiy, Disc, ExpVXY, ExpVXYZ, ExpX, fExpXY, NSfExpXY, SinPix, and Constant.

7.26.3 Member Data Documentation

7.26.3.1 int FunctionV::dim [protected]

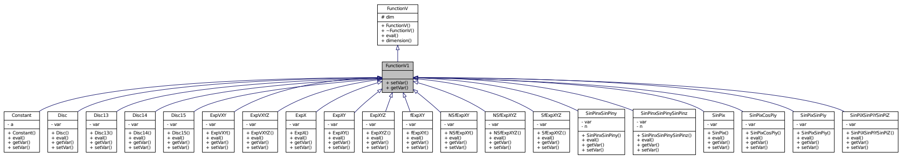
The documentation for this class was generated from the following file:

- FunctionV.hpp

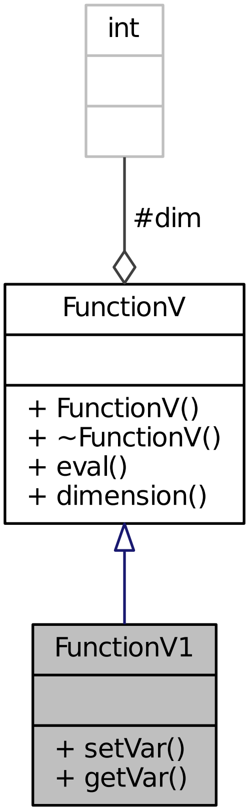
7.27 FunctionV1 Class Reference

```
#include <FunctionV1.hpp>
```

Inheritance diagram for FunctionV1:



Collaboration diagram for FunctionV1:



Public Member Functions

- virtual void setVar (ldouble x)=0
- virtual ldouble getVar (void)=0

Additional Inherited Members

7.27.1 Member Function Documentation

7.27.1.1 virtual **ldouble** FunctionV1::getVar (void) [pure virtual]

Implemented in Disc13, Disc15, Disc, Disc14, ExpVXY, ExpVXYZ, NSfExpXYZ, SfExpXYZ, SinPinxSinPinySinPinz, SinPiXSinPiYSinPiZ, ExpXY, ExpXYZ, fExpXY, NSfExpXY, SinPinxSinPiny, SinPixCosPiy, SinPixSinPiy, ExpX, SinPix, and Constant.

7.27.1.2 virtual void FunctionV1::setVar (Idouble *x*) [pure virtual]

Implemented in Disc13, Disc15, Disc, Disc14, ExpVXY, ExpVXYZ, NSfExpXYZ, SfExpXYZ, SinPinxSinPinySinPinz, SinPiXSinPiYSinPiZ, ExpXY, ExpXYZ, fExpXY, NSfExpXY, SinPinxSinPiny, SinPixCosPiy, SinPixSinPiy, ExpX, SinPix, and Constant.

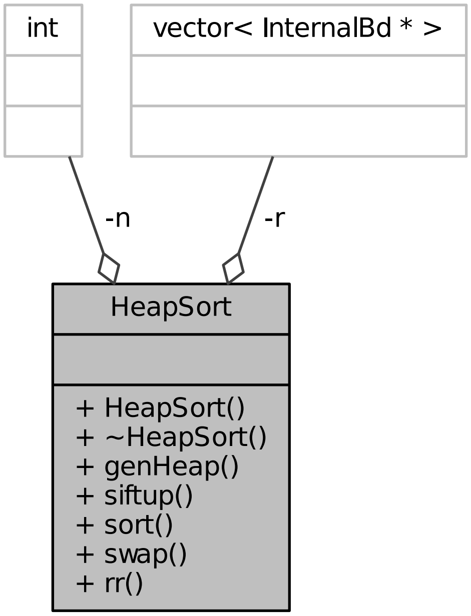
The documentation for this class was generated from the following file:

- FunctionV1.hpp

7.28 HeapSort Class Reference

`#include <HeapSort.hpp>`

Collaboration diagram for HeapSort:



Public Member Functions

- HeapSort (vector< InternalBd ∗ > &a, int n)
- ∼HeapSort ()
- void genHeap (void)
- void siftup (int i, int n)
- void sort (void)
- void swap (int i, int j)
- InternalBd ∗ rr (int i)

Private Attributes

- vector< InternalBd * > r
- int n

7.28.1 Constructor & Destructor Documentation

7.28.1.1 HeapSort::HeapSort (vector< InternalBd * > & *a*, int *n*) [inline]

7.28.1.2 HeapSort::~HeapSort ( ) [inline]

7.28.2 Member Function Documentation

7.28.2.1 void HeapSort::genHeap (void) [inline]

7.28.2.2 InternalBd* HeapSort::rr (int *i*) [inline]

7.28.2.3 void HeapSort::siftup (int *i*, int *n*) [inline]

7.28.2.4 void HeapSort::sort (void) [inline]

7.28.2.5 void HeapSort::swap (int *i*, int *j*) [inline]

7.28.3 Member Data Documentation

7.28.3.1 int HeapSort::n [private]

7.28.3.2 vector<InternalBd*> HeapSort::r [private]

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

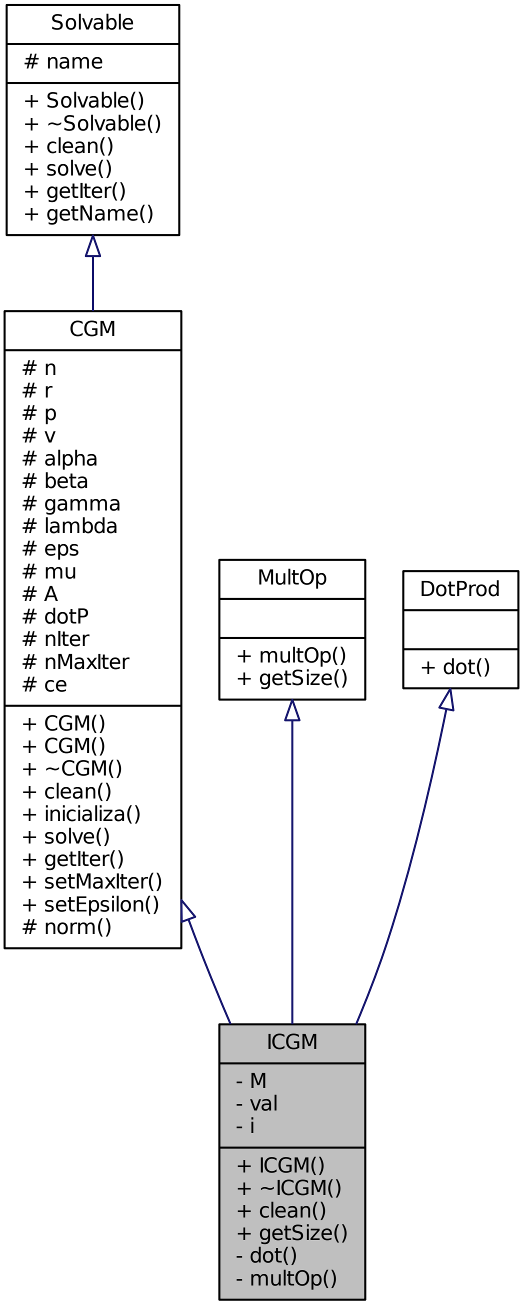
- HeapSort.hpp

7.29 ICGM Class Reference

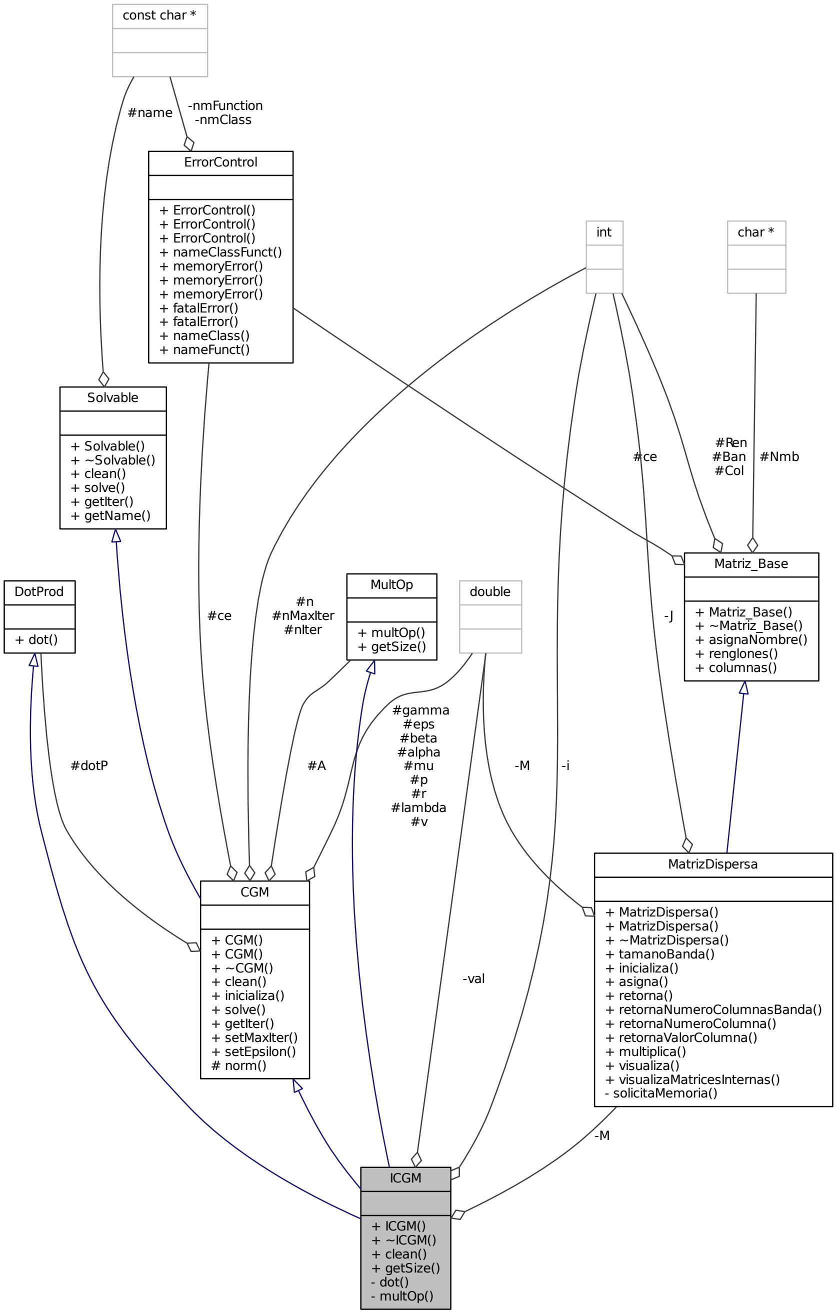
Clase para implementar CGM con matrices bandadas o dispersas.

```
#include <ICGM.hpp>
```

Inheritance diagram for ICGM:



Collaboration diagram for ICGM:



Public Member Functions

- ICGM (int n, MatrizDispersa ∗M, ldouble eps, int iter)

  Contructor de la clase.
- ∼ICGM ()

Destructor de la clase.

- void clean (void)
- int getSize (void)

    vector size

Private Member Functions

- ldouble dot (ldouble ∗u, ldouble ∗v)

    Producto punto.
- void multOp (ldouble ∗u, ldouble ∗v)

    Multiplica Au=v.

Private Attributes

- MatrizDispersa ∗ M

    Multiplica Au=v.
- ldouble val

    Variables temporales.
- int i

Additional Inherited Members

7.29.1 Detailed Description

Clase para implementar CGM con matrices bandadas o dispersas.

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.1

Bug No hay errores conocidos

7.29.2 Constructor & Destructor Documentation

7.29.2.1 ICGM::ICGM (int *n,* MatrizDispersa ∗ *M,* ldouble *eps,* int *iter*) [inline]

Contructor de la clase.

7.29.2.2 ICGM::∼ICGM ( ) [inline]

Destructor de la clase.

7.29.3 Member Function Documentation

7.29.3.1 void ICGM::clean (void) [inline], [virtual]

Reimplemented from CGM.

7.29.3.2 ldouble ICGM::dot (ldouble ∗ *u*, ldouble ∗ *v*) [inline], [private], [virtual]

Producto punto.

Implements DotProd.

7.29.3.3 int ICGM::getSize (void) [inline], [virtual]

vector size

Implements MultOp.

7.29.3.4 void ICGM::multOp (ldouble ∗ *u*, ldouble ∗ *v*) [inline], [private], [virtual]

Multiplica Au=v.

Implements MultOp.

7.29.4 Member Data Documentation

7.29.4.1 int ICGM::i [private]

7.29.4.2 MatrizDispersa∗ ICGM::M [private]

Multiplica Au=v.

7.29.4.3 ldouble ICGM::val [private]

Variables temporales.

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

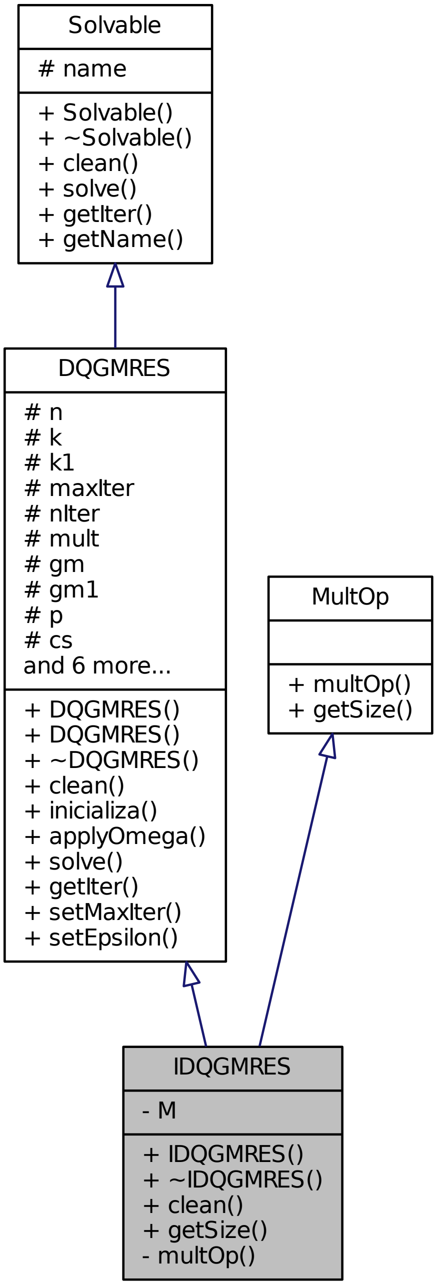
- ICGM.hpp

7.30 IDQGMRES Class Reference

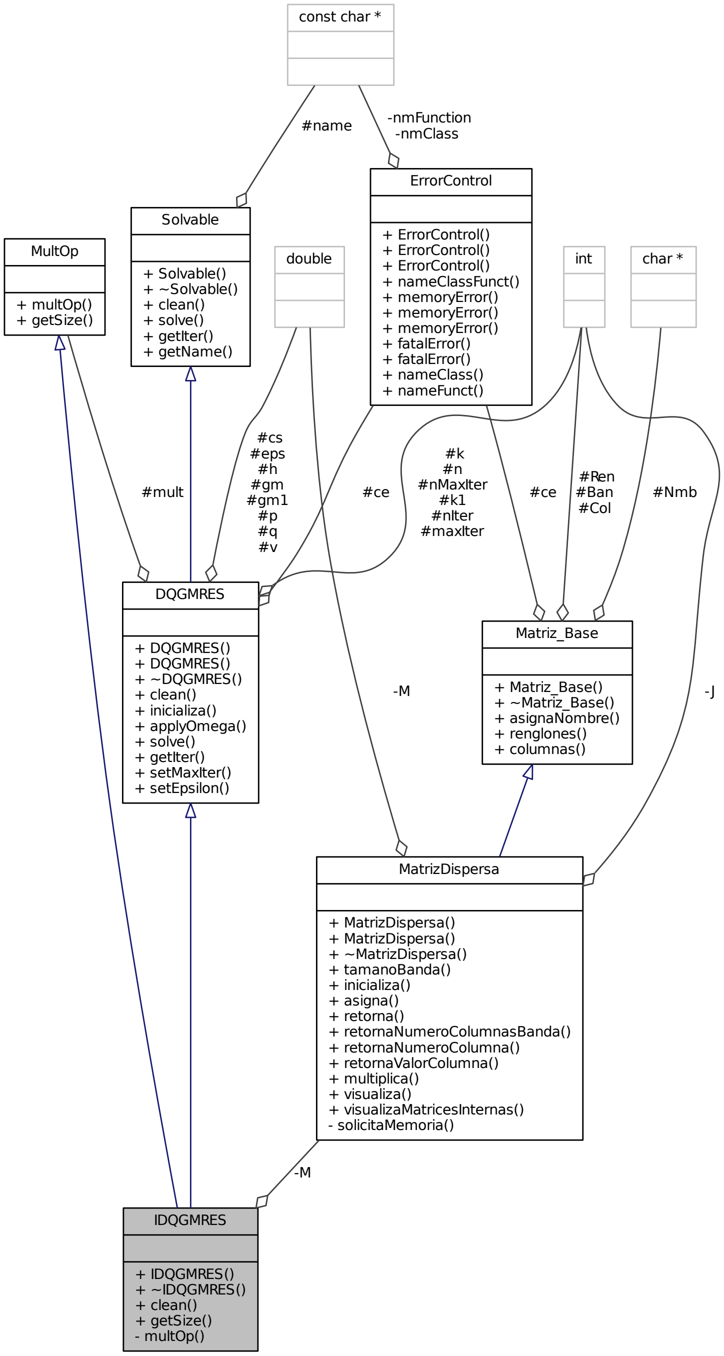
Clase para implementar DQGMRES con matrices bandadas o dispersas.

```
#include <IDQGMRES.hpp>
```

Inheritance diagram for IDQGMRES:



Collaboration diagram for IDQGMRES:



Public Member Functions

- IDQGMRES (int n, MatrizDispersa ∗M, int k, double eps, int iter)

Constructor de la clase.
- ∼IDQGMRES ()
- void clean (void)
- int getSize (void)

  vector size

Private Member Functions

- void multOp (ldouble ∗u, ldouble ∗v)

  Multiplica Au=v.

Private Attributes

- MatrizDispersa ∗ M

  Matriz Bandada o Dispersa.

Additional Inherited Members

7.30.1 Detailed Description

Clase para implementar DQGMRES con matrices bandadas o dispersas.

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.1

Bug No hay errores conocidos

7.30.2 Constructor & Destructor Documentation

7.30.2.1 IDQGMRES::IDQGMRES (int *n*, MatrizDispersa ∗ *M*, int *k*, double *eps*, int *iter*) `[inline]`

Constructor de la clase.

7.30.2.2 IDQGMRES::∼IDQGMRES ( ) `[inline]`

7.30.3 Member Function Documentation

7.30.3.1 void IDQGMRES::clean (void) `[inline]`, `[virtual]`

Reimplemented from DQGMRES.

7.30.3.2 int IDQGMRES::getSize (void) [inline], [virtual]

vector size

Implements MultOp.

7.30.3.3 void IDQGMRES::multOp (Idouble ∗ *u*, Idouble ∗ *v*) [inline], [private], [virtual]

Multiplica Au=v.

Implements MultOp.

7.30.4 Member Data Documentation

7.30.4.1 MatrizDispersa∗ IDQGMRES::M [private]

Matriz Bandada o Dispersa.

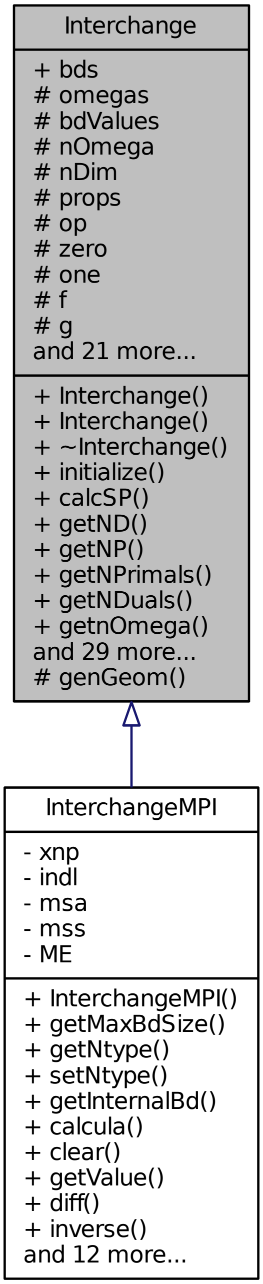
The documentation for this class was generated from the following file:

- IDQGMRES.hpp

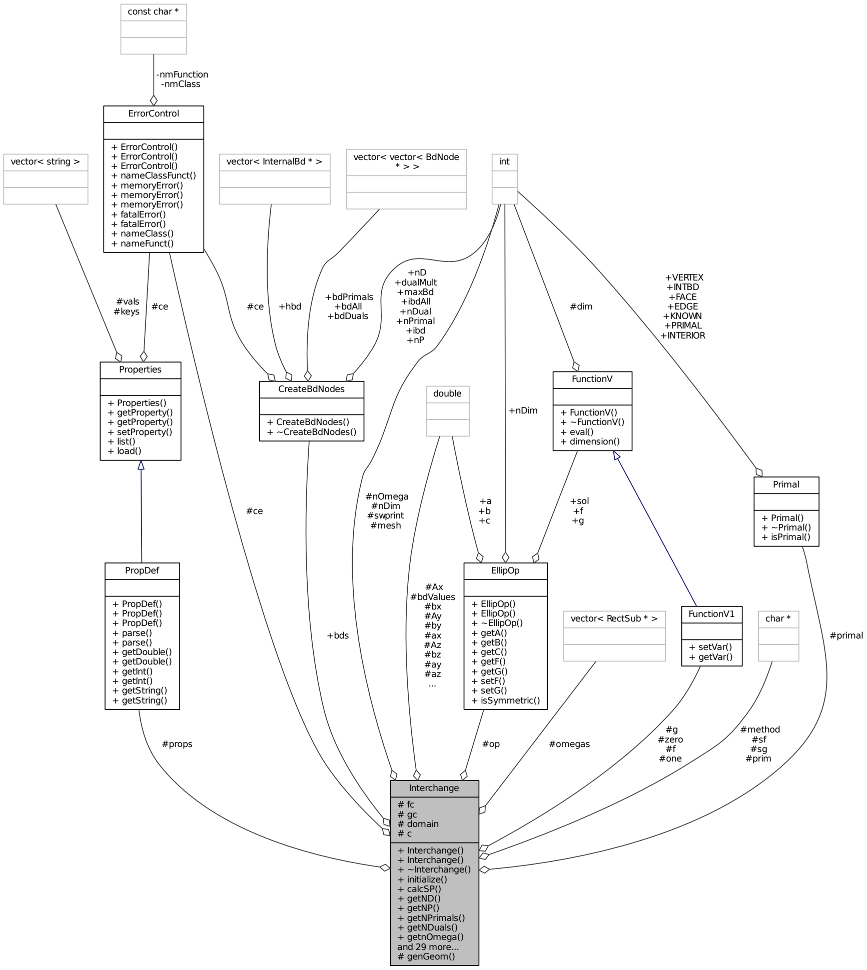
7.31 Interchange Class Reference

```
#include <Interchange.hpp>
```

Inheritance diagram for Interchange:



Collaboration diagram for Interchange:



Public Member Functions

- Interchange (PropDef &props)

    Constructor.
- Interchange (void)

    Constructor.
- virtual ∼Interchange ()

    Destructor.
- void initialize (int nOmega)
- Solvable ∗ calcSP (int sp)
- int getND (void)
- int getNP (void)
- int getNPrimals (void)
- int getNDuals (void)
- int getnOmega (void)
- int getnDim (void)

- ldouble rbdValues (int i, int j)
- void sbdValues (int i, int j, ldouble v)
- void pbdValues (void)
- void clear (int e, int sc)

    Clear scr[sc][] en e subdomains.
- void setValue (int e, int sc, int n, ldouble val)
- void inverse (int e, int sp, int sc1, int sc2)

    scr[sc2][] = A(sp)-1(scr[sc1][])
- void multOp (int e, int sc1, int sc2)

    scr[s2][] = A(scr[sc1][ ])
- virtual void calcula (int e, int node, int sp)
- virtual void clear (int sc)

    Clear scr[sc][] in all subdomains.
- virtual ldouble getValue (int e, int scr, int node)
- virtual ldouble getValue (int e, int scr1, int scr2, int node)
- virtual void diff (int sc3, int sc1, int sc2)

    scr[sc3][] = scr[sc1][] - scr[sc2][] in all subdomains
- virtual void inverse (int sp, int sc1, int sc2)

    scr[sc2][] = A(sp)-1(scr[sc1][])
- virtual void knownValues (int sc)

    scr[sc][] = Dirichlet boundary values of all subdomains
- virtual void multOp (int sc1, int sc2)

    scr[s2][] = A(scr[sc1][ ])
- virtual void rhs (int sc)

    scr[sc][] = initial right-hand-side (all subdomains)
- virtual void genInv (int e, int type)
- virtual void getCoordNode (int e, int n, ldouble ∗x)
- virtual void print (const char ∗s, int sc)
- virtual void print (int sc)
- virtual int getMaxBdSize (void)
- virtual int ∗ getNtype (int e)
- virtual void setNtype (int e, int ∗arr)
- virtual void diffValues (int sc)

    bdValues[][] -= scr[sc][] in all subdomains
- virtual void fromSubdomains (int sc)

    bdValues[][] = scr[sc][] from all subdomains
- virtual void getPrimals (int sc)

    bdValues[][] (primals only) = scr[sc][] (primals)
- virtual void setPrimals (int sc)

    scr[sc][] = bdValues all subdomains
- virtual void toSubdomains (int sc)

    scr[sc][] = bdValues[][] all subdomains

Public Attributes

- CreateBdNodes ∗ bds

Protected Member Functions

- void genGeom (void)

Protected Attributes

- vector$<$ RectSub $*$ $>$ omegas
- ldouble $**$ bdValues
- int nOmega
- int nDim
- PropDef $*$ props
- EllipOp $*$ op
- FunctionV1 $*$ zero
- FunctionV1 $*$ one
- FunctionV1 $*$ f
- FunctionV1 $*$ g
- char $*$ sf
- char $*$ sg
- ldouble fc
- ldouble gc
- int $*$ mesh
- char $*$ prim
- char $*$ method
- int swprint
- ldouble Ax
- ldouble Ay
- ldouble Az
- ldouble $**$ domain
- ldouble ax
- ldouble ay
- ldouble az
- ldouble c
- ldouble bx
- ldouble by
- ldouble bz
- Primal $*$ primal
- ErrorControl ce

  Control de errores.

7.31.1 Constructor & Destructor Documentation

7.31.1.1 Interchange::Interchange (PropDef & *props*)

Constructor.

7.31.1.2 Interchange::Interchange (void) [inline]

Constructor.

7.31.1.3 virtual Interchange::∼Interchange ( ) [inline], [virtual]

Destructor.

7.31.2 Member Function Documentation

7.31.2.1 Solvable ∗ Interchange::calcSP (int *sp*)

7.31.2.2 virtual void Interchange::calcula (int *e,* int *node,* int *sp*) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.3 void Interchange::clear (int *e,* int *sc*) [inline]

Clear scr[sc][] en e subdomains.

7.31.2.4 virtual void Interchange::clear (int *sc*) [inline], [virtual]

Clear scr[sc][] in all subdomains.

Reimplemented in InterchangeMPI.

7.31.2.5 virtual void Interchange::diff (int *sc3,* int *sc1,* int *sc2*) [inline], [virtual]

scr[sc3][] = scr[sc1][] - scr[sc2][] in all subdomains

Reimplemented in InterchangeMPI.

7.31.2.6 virtual void Interchange::diffValues (int *sc*) [inline], [virtual]

bdValues[][] -= scr[sc][] in all subdomains

Reimplemented in InterchangeMPI.

7.31.2.7 virtual void Interchange::fromSubdomains (int *sc*) [inline], [virtual]

bdValues[][] = scr[sc][] from all subdomains

Reimplemented in InterchangeMPI.

7.31.2.8 void Interchange::genGeom (void) [protected]

7.31.2.9 virtual void Interchange::genInv (int *e,* int *type*) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.10 virtual void Interchange::getCoordNode (int *e,* int *n,* ldouble ∗ *x*) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.11 virtual int Interchange::getMaxBdSize (void) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.12 int Interchange::getND (void) [inline]

7.31.2.13 int Interchange::getnDim (void) [inline]

7.31.2.14 int Interchange::getNDuals (void) [inline]

7.31.2.15 int Interchange::getnOmega (void) [inline]

7.31.2.16 int Interchange::getNP (void) [inline]

7.31.2.17 int Interchange::getNPrimals (void) [inline]

7.31.2.18 virtual int∗ Interchange::getNtype (int *e*) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.19 virtual void Interchange::getPrimals (int *sc*) [inline], [virtual]

bdValues[][] (primals only) = scr[sc][] (primals)

Reimplemented in InterchangeMPI.

7.31.2.20 virtual ldouble Interchange::getValue (int *e,* int *scr,* int *node*) [inline], [virtual]

7.31.2.21 virtual ldouble Interchange::getValue (int *e,* int *scr1,* int *scr2,* int *node*) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.22 void Interchange::initialize (int *nOmega*)

7.31.2.23 void Interchange::inverse (int *e,* int *sp,* int *sc1,* int *sc2*) [inline]

scr[sc2][] = A(sp)-1(scr[sc1][])

7.31.2.24 virtual void Interchange::inverse (int *sp,* int *sc1,* int *sc2*) [inline], [virtual]

scr[sc2][] = A(sp)-1(scr[sc1][])

Reimplemented in InterchangeMPI.

7.31.2.25 virtual void Interchange::knownValues (int *sc*) [inline], [virtual]

scr[sc][] = Dirichlet boundary values of all subdomains

Reimplemented in InterchangeMPI.

7.31.2.26 void Interchange::multOp (int *e,* int *sc1,* int *sc2*) [inline]

scr[s2][] = A(scr[sc1][ ])

7.31.2.27 virtual void Interchange::multOp (int *sc1,* int *sc2*) [inline], [virtual]

scr[s2][] = A(scr[sc1][ ])

Reimplemented in InterchangeMPI.

7.31.2.28 void Interchange::pbdValues (void) [inline]

7.31.2.29 virtual void Interchange::print (const char ∗ *s,* int *sc*) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.30 virtual void Interchange::print (int *sc*) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.31 ldouble Interchange::rbdValues (int *i,* int *j*) [inline]

7.31.2.32 virtual void Interchange::rhs (int *sc*) [inline], [virtual]

scr[sc][] = initial right-hand-side (all subdomains)

Reimplemented in InterchangeMPI.

7.31.2.33 void Interchange::sbdValues (int *i,* int *j,* ldouble *v*) [inline]

7.31.2.34 virtual void Interchange::setNtype (int *e,* int ∗ *arr*) [inline], [virtual]

Reimplemented in InterchangeMPI.

7.31.2.35 virtual void Interchange::setPrimals (int *sc*) [inline], [virtual]

scr[sc][] = bdValues all subdomains

Reimplemented in InterchangeMPI.

7.31.2.36 void Interchange::setValue (int *e,* int *sc,* int *n,* ldouble *val*) [inline]

7.31.2.37 virtual void Interchange::toSubdomains (int *sc*) [inline], [virtual]

scr[sc][] = bdValues[][] all subdomains

Reimplemented in InterchangeMPI.

7.31.3 Member Data Documentation

7.31.3.1 Idouble Interchange::Ax [protected]

7.31.3.2 Idouble Interchange::ax [protected]

7.31.3.3 Idouble Interchange::Ay [protected]

7.31.3.4 Idouble Interchange::ay [protected]

7.31.3.5 Idouble Interchange::Az [protected]

7.31.3.6 Idouble Interchange::az [protected]

7.31.3.7 CreateBdNodes* Interchange::bds

7.31.3.8 Idouble** Interchange::bdValues [protected]

7.31.3.9 Idouble Interchange::bx [protected]

7.31.3.10 Idouble Interchange::by [protected]

7.31.3.11 Idouble Interchange::bz [protected]

7.31.3.12 Idouble Interchange::c [protected]

7.31.3.13 ErrorControl Interchange::ce [protected]

Control de errores.

7.31.3.14 Idouble** Interchange::domain [protected]

7.31.3.15 FunctionV1* Interchange::f [protected]

7.31.3.16 Idouble Interchange::fc [protected]

7.31.3.17 FunctionV1 * Interchange::g [protected]

7.31.3.18 Idouble Interchange::gc [protected]

7.31.3.19 int* Interchange::mesh [protected]

7.31.3.20 char* Interchange::method [protected]

7.31.3.21 int Interchange::nDim [protected]

7.31.3.22 int Interchange::nOmega [protected]

7.31.3.23 vector<RectSub*> Interchange::omegas [protected]

7.31.3.24 FunctionV1 * Interchange::one [protected]

7.31.3.25 EllipOp∗ Interchange::op [protected]

7.31.3.26 char∗ Interchange::prim [protected]

7.31.3.27 Primal∗ Interchange::primal [protected]

7.31.3.28 PropDef∗ Interchange::props [protected]

7.31.3.29 char∗ Interchange::sf [protected]

7.31.3.30 char ∗ Interchange::sg [protected]

7.31.3.31 int Interchange::swprint [protected]

7.31.3.32 FunctionV1∗ Interchange::zero [protected]

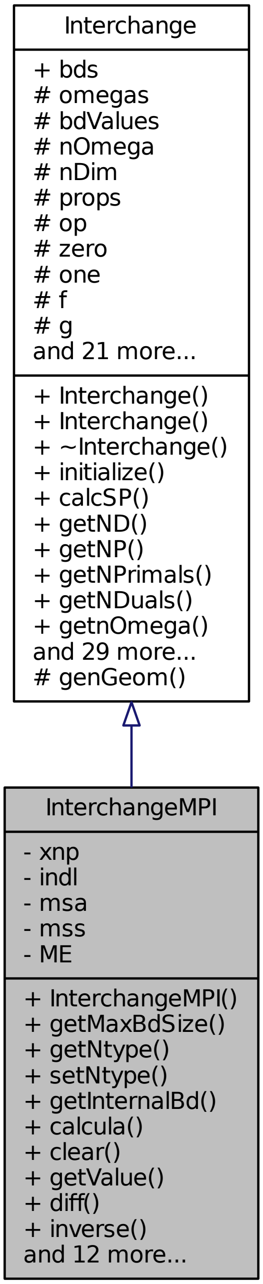
The documentation for this class was generated from the following files:

- Interchange.hpp
- Interchange.cpp

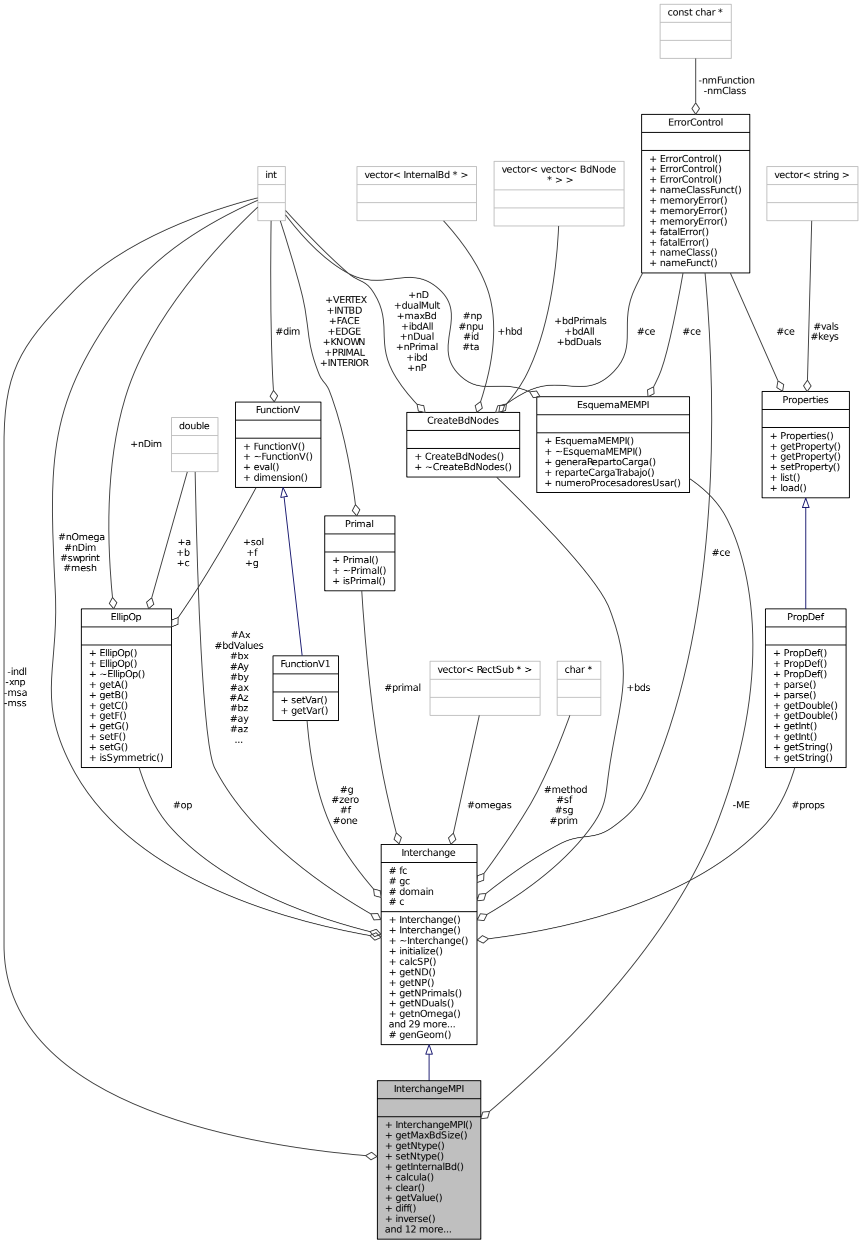
7.32 InterchangeMPI Class Reference

```
#include <InterchangeMPI.hpp>
```

Inheritance diagram for InterchangeMPI:



Collaboration diagram for InterchangeMPI:



Public Member Functions

- InterchangeMPI (PropDef &props, EsquemaMEMPI &me)

  Constructor.
- int getMaxBdSize (void)
- int $*$ getNtype (int e)
- void setNtype (int e, int $*$arr)
- vector$<$ InternalBd $*$ $>$ getInternalBd (int e)

- void calcula (int e, int node, int sp)
- void clear (int sc)

    Clear scr[sc][] in all subdomains.
- ldouble getValue (int e, int scr1, int scr2, int node)
- void diff (int sc3, int sc1, int sc2)

    scr[sc3][] = scr[sc1][] - scr[sc2][] in all subdomains
- void inverse (int sp, int sc1, int sc2)

    scr[sc2][] = A(sp)-1(scr[sc1][])
- void knownValues (int sc)

    scr[sc][] = Dirichlet boundary values of all subdomains
- void multOp (int sc1, int sc2)

    scr[s2][] = A(scr[sc1][ ])
- void rhs (int sc)

    scr[sc][] = initial right-hand-side (all subdomains)
- void genInv (int e, int type)
- void getCoordNode (int e, int n, ldouble ∗x)
- void print (const char ∗s, int sc)
- void print (int sc)
- void diffValues (int sc)

    bdValues[][] -= scr[sc][] in all subdomains
- void fromSubdomains (int sc)

    bdValues[][] = scr[sc][] from all subdomains
- void getPrimals (int sc)

    bdValues[][] (primals only) = scr[sc][] (primals)
- void setPrimals (int sc)

    scr[sc][] = bdValues all subdomains
- void toSubdomains (int sc)

    scr[sc][] = bdValues[][] all subdomains

Private Attributes

- int xnp

    Numero de esclavo en el que estara la tarea.
- int indl

    Numero de tarea dentro del esclavo.
- int msa [10]

    Arreglo para recibir mensajes.
- int mss [10]

    Arreglo para enviar mensajes.
- EsquemaMEMPI ∗ ME

    Puntero al esquema Maestro-Esclavo.

Additional Inherited Members

7.32.1 Constructor & Destructor Documentation

7.32.1.1 InterchangeMPI::InterchangeMPI (PropDef & *props,* EsquemaMEMPI & *me*)

Constructor.

7.32.2 Member Function Documentation

7.32.2.1 void InterchangeMPI::calcula (int *e*, int *node*, int *sp*) [virtual]

Reimplemented from Interchange.

7.32.2.2 void InterchangeMPI::clear (int *sc*) [virtual]

Clear scr[sc][] in all subdomains.

Reimplemented from Interchange.

7.32.2.3 void InterchangeMPI::diff (int *sc3*, int *sc1*, int *sc2*) [virtual]

scr[sc3][] = scr[sc1][] - scr[sc2][] in all subdomains

Reimplemented from Interchange.

7.32.2.4 void InterchangeMPI::diffValues (int *sc*) [virtual]

bdValues[][] -= scr[sc][] in all subdomains

Reimplemented from Interchange.

7.32.2.5 void InterchangeMPI::fromSubdomains (int *sc*) [virtual]

bdValues[][] = scr[sc][] from all subdomains

Reimplemented from Interchange.

7.32.2.6 void InterchangeMPI::genInv (int *e*, int *type*) [virtual]

Reimplemented from Interchange.

7.32.2.7 void InterchangeMPI::getCoordNode (int *e*, int *n*, ldouble ∗ *x*) [virtual]

Reimplemented from Interchange.

7.32.2.8 vector< InternalBd ∗ > InterchangeMPI::getInternalBd (int *e*)

7.32.2.9 int InterchangeMPI::getMaxBdSize (void) [virtual]

Reimplemented from Interchange.

7.32.2.10 int ∗ InterchangeMPI::getNtype (int *e*) [virtual]

Reimplemented from Interchange.

7.32.2.11 void InterchangeMPI::getPrimals (int *sc*) [virtual]

bdValues[][] (primals only) = scr[sc][] (primals)

Reimplemented from Interchange.

7.32.2.12 ldouble InterchangeMPI::getValue (int *e*, int *scr1*, int *scr2*, int *node*) [virtual]

Reimplemented from Interchange.

7.32.2.13 void InterchangeMPI::inverse (int *sp*, int *sc1*, int *sc2*) [virtual]

scr[sc2][] = A(sp)-1(scr[sc1][])

Reimplemented from Interchange.

7.32.2.14 void InterchangeMPI::knownValues (int *sc*) [virtual]

scr[sc][] = Dirichlet boundary values of all subdomains

Reimplemented from Interchange.

7.32.2.15 void InterchangeMPI::multOp (int *sc1*, int *sc2*) [virtual]

scr[s2][] = A(scr[sc1][ ])

Reimplemented from Interchange.

7.32.2.16 void InterchangeMPI::print (const char * *s*, int *sc*) [virtual]

Reimplemented from Interchange.

7.32.2.17 void InterchangeMPI::print (int *sc*) [virtual]

Reimplemented from Interchange.

7.32.2.18 void InterchangeMPI::rhs (int *sc*) [virtual]

scr[sc][] = initial right-hand-side (all subdomains)

Reimplemented from Interchange.

7.32.2.19 void InterchangeMPI::setNtype (int *e*, int * *arr*) [virtual]

Reimplemented from Interchange.

7.32.2.20 void InterchangeMPI::setPrimals (int *sc*) [virtual]

scr[sc][] = bdValues all subdomains

Reimplemented from Interchange.

7.32.2.21 void InterchangeMPI::toSubdomains (int *sc*) [virtual]

scr[sc][] = bdValues[][] all subdomains

Reimplemented from Interchange.

7.32.3 Member Data Documentation

7.32.3.1 int InterchangeMPI::indl [private]

Numero de tarea dentro del esclavo.

7.32.3.2 EsquemaMEMPI* InterchangeMPI::ME [private]

Puntero al esquema Maestro-Esclavo.

7.32.3.3 int InterchangeMPI::msa[10] [private]

Arreglo para recibir mensajes.

7.32.3.4 int InterchangeMPI::mss[10] [private]

Arreglo para enviar mensajes.

7.32.3.5 int InterchangeMPI::xnp [private]

Numero de esclavo en el que estara la tarea.

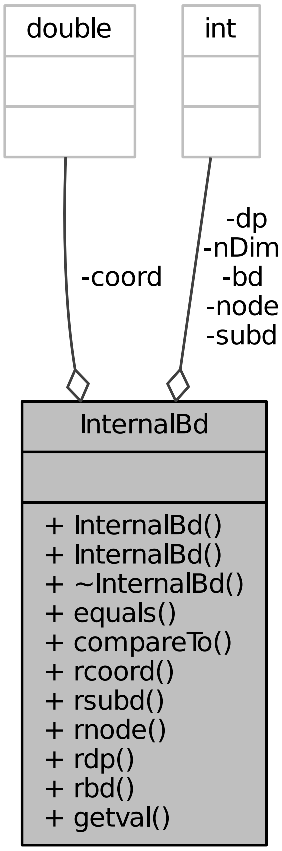
The documentation for this class was generated from the following files:

- InterchangeMPI.hpp
- InterchangeMPI.cpp

7.33 InternalBd Class Reference

```
#include <InternalBd.hpp>
```

Collaboration diagram for InternalBd:



Public Member Functions

- InternalBd (void)
- InternalBd (int s, int n, int b, int i, int d, ldouble ∗cor)
- ∼InternalBd (void)
- bool equals (InternalBd ∗x)
- int compareTo (InternalBd ∗a)
- ldouble rcoord (int i)
- int rsubd (void)
- int rnode (void)
- int rdp (void)
- int rbd (void)
- void getval (int &s, int &n, int &b, int &i, int &d, ldouble ∗c)

Private Attributes

- int subd
- int node
- int bd
- int dp
- int nDim
- ldouble ∗ coord

7.33.1 Constructor & Destructor Documentation

7.33.1.1 InternalBd::InternalBd (void) [inline]

7.33.1.2 InternalBd::InternalBd (int *s*, int *n*, int *b*, int *i*, int *d*, ldouble ∗ *cor*) [inline]

7.33.1.3 InternalBd::∼InternalBd (void) [inline]

7.33.2 Member Function Documentation

7.33.2.1 int InternalBd::compareTo (InternalBd ∗ *a*) [inline]

7.33.2.2 bool InternalBd::equals (InternalBd ∗ *x*) [inline]

7.33.2.3 void InternalBd::getval (int & *s*, int & *n*, int & *b*, int & *i*, int & *d*, ldouble ∗ *c*) [inline]

7.33.2.4 int InternalBd::rbd (void) [inline]

7.33.2.5 ldouble InternalBd::rcoord (int *i*) [inline]

7.33.2.6 int InternalBd::rdp (void) [inline]

7.33.2.7 int InternalBd::rnode (void) [inline]

7.33.2.8 int InternalBd::rsubd (void) [inline]

7.33.3 Member Data Documentation

7.33.3.1 int InternalBd::bd [private]

7.33.3.2 ldouble∗ InternalBd::coord [private]

7.33.3.3 int InternalBd::dp [private]

7.33.3.4 int InternalBd::nDim [private]

7.33.3.5 int InternalBd::node [private]

7.33.3.6 int InternalBd::subd [private]

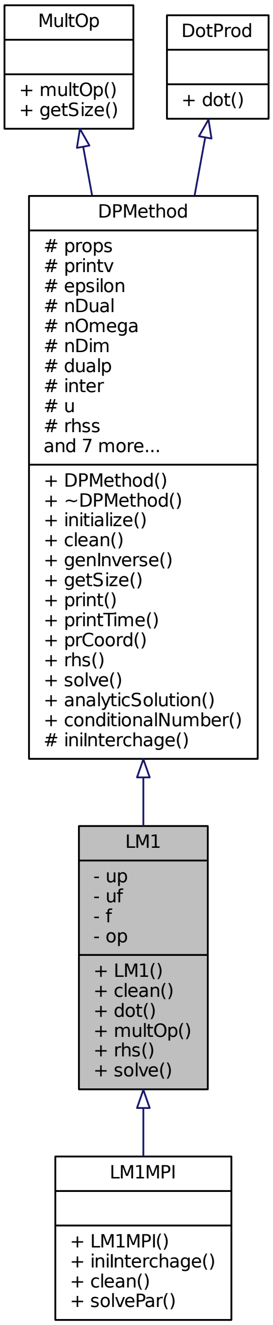
The documentation for this class was generated from the following file:

- InternalBd.hpp

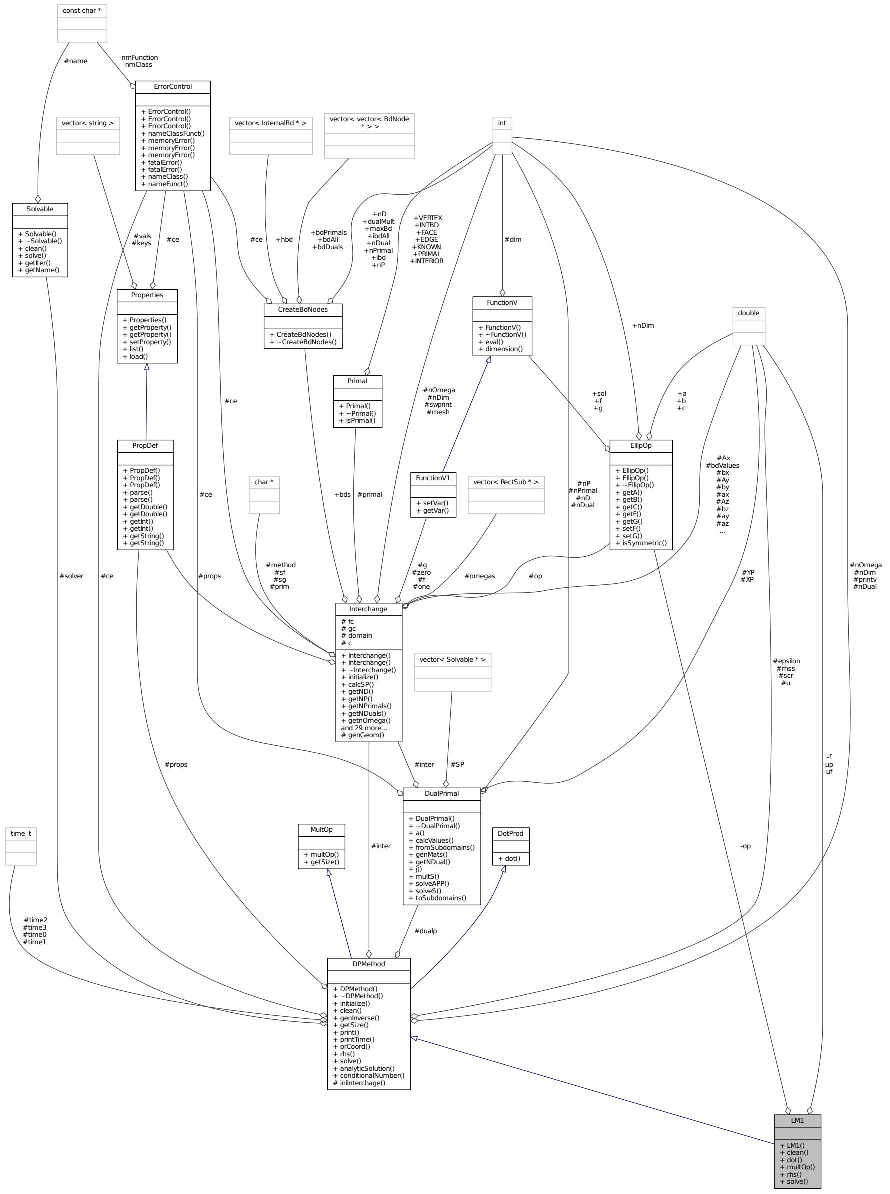
7.34 LM1 Class Reference

```
#include <LM1.hpp>
```

Inheritance diagram for LM1:



Collaboration diagram for LM1:



Public Member Functions

- LM1 (PropDef &props, EllipOp &op)
- virtual void clean (void)
- ldouble dot (ldouble ∗u, ldouble ∗v)
- void multOp (ldouble ∗u, ldouble ∗v)

  y = A∗x
- void rhs (void)
- void solve (void)

Private Attributes

- ldouble ∗ up
- ldouble ∗ uf
- ldouble ∗ f
- EllipOp ∗ op

Additional Inherited Members

7.34.1 Constructor & Destructor Documentation

7.34.1.1 LM1::LM1 (PropDef & *props*, EllipOp & *op*) [inline]

7.34.2 Member Function Documentation

7.34.2.1 virtual void LM1::clean (void) [inline], [virtual]

Implements DPMethod.

Reimplemented in LM1MPI.

7.34.2.2 ldouble LM1::dot (ldouble ∗ *u*, ldouble ∗ *v*) [virtual]

Implements DotProd.

7.34.2.3 void LM1::multOp (ldouble ∗ *x*, ldouble ∗ *y*) [virtual]

y = A∗x

Implements MultOp.

7.34.2.4 void LM1::rhs (void) [virtual]

Implements DPMethod.

7.34.2.5 void LM1::solve (void) [virtual]

Implements DPMethod.

7.34.3 Member Data Documentation

7.34.3.1 ldouble∗ LM1::f [private]

7.34.3.2 EllipOp∗ LM1::op [private]

7.34.3.3 ldouble∗ LM1::uf [private]

7.34.3.4 ldouble∗ LM1::up [private]

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

- LM1.hpp

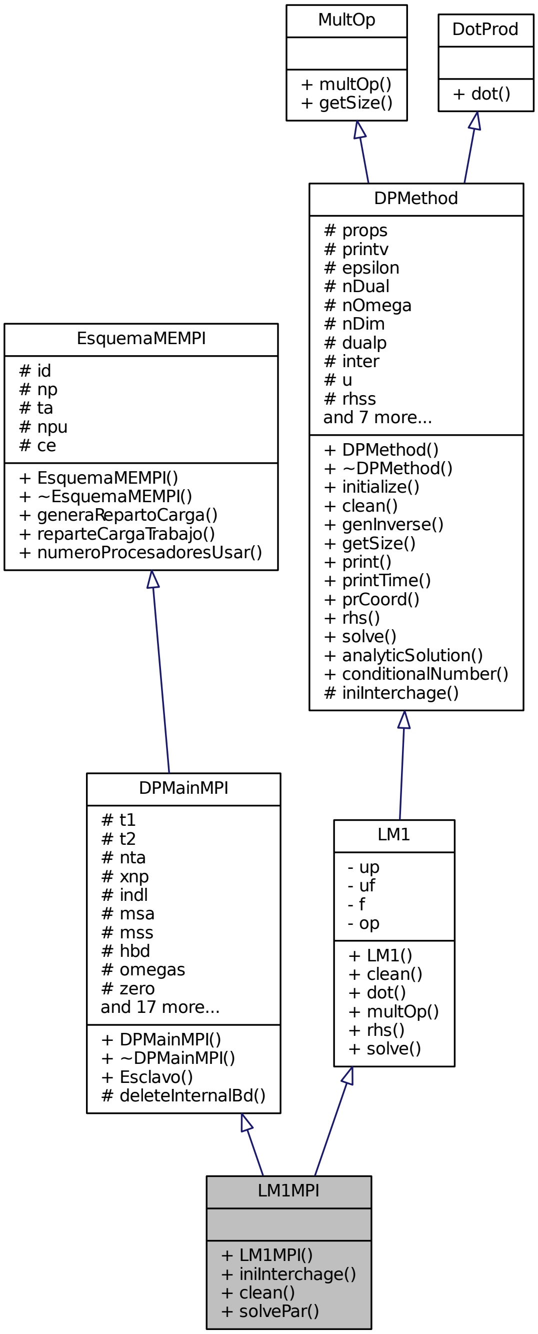
- LM1.cpp

7.35 LM1MPI Class Reference

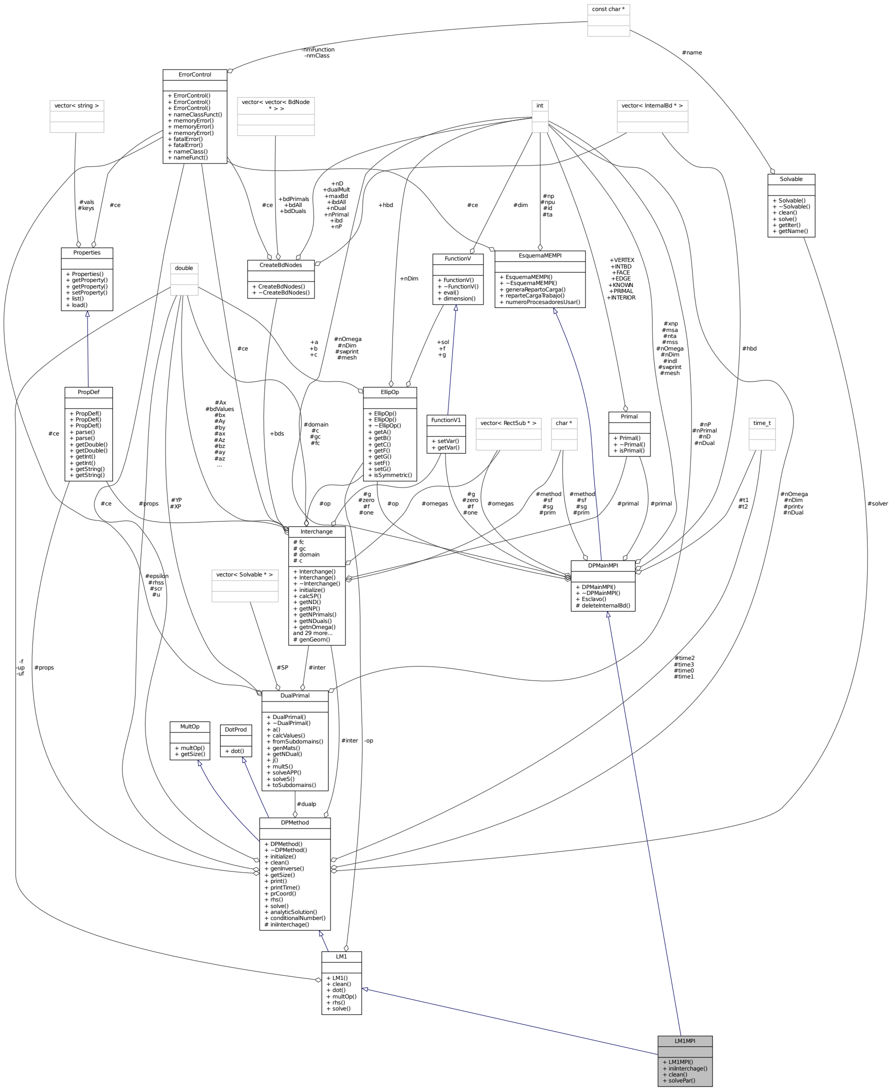
Clase para definir el metodo LM-1 de DVS-DDM.

```
#include <LM1MPI.hpp>
```

Inheritance diagram for LM1MPI:



Collaboration diagram for LM1MPI:



Public Member Functions

- LM1MPI (int id, int np, PropDef &props, EllipOp &op)

  Constructor de la clase.
- void iniInterchage (void)

  Inicializa InterchangeMPI en lugar de Interchange.
- void clean (void)
- void solvePar (void)

  Sobrecarga del la aplicacion.

Additional Inherited Members

7.35.1 Detailed Description

Clase para definir el metodo LM-1 de DVS-DDM.

Clase para definir el metodo LM-1 de DVS-DDM en paralelo

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.35.2 Constructor & Destructor Documentation

7.35.2.1 LM1MPI::LM1MPI (int *id*, int *np*, PropDef & *props*, EllipOp & *op*) [inline]

Constructor de la clase.

7.35.3 Member Function Documentation

7.35.3.1 void LM1MPI::clean (void) [inline], [virtual]

Reimplemented from LM1.

7.35.3.2 void LM1MPI::iniInterchage (void) [inline], [virtual]

Inicializa InterchangeMPI en lugar de Interchange.

Reimplemented from DPMethod.

7.35.3.3 void LM1MPI::solvePar (void) [inline]

Sobrecarga del la aplicacion.

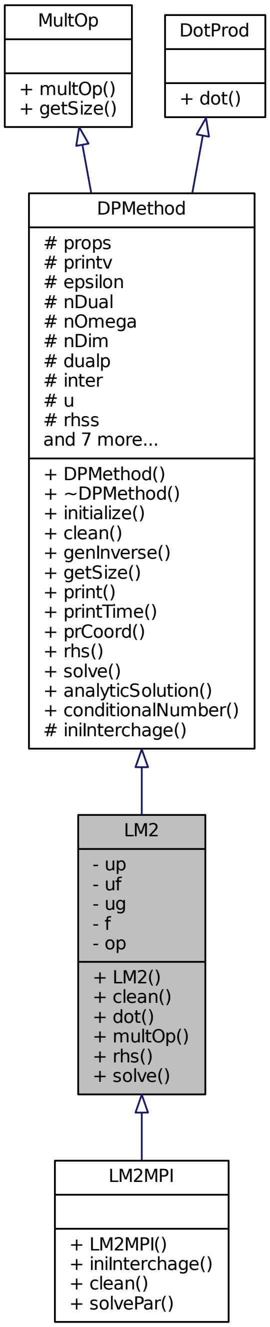
The documentation for this class was generated from the following file:

- LM1MPI.hpp

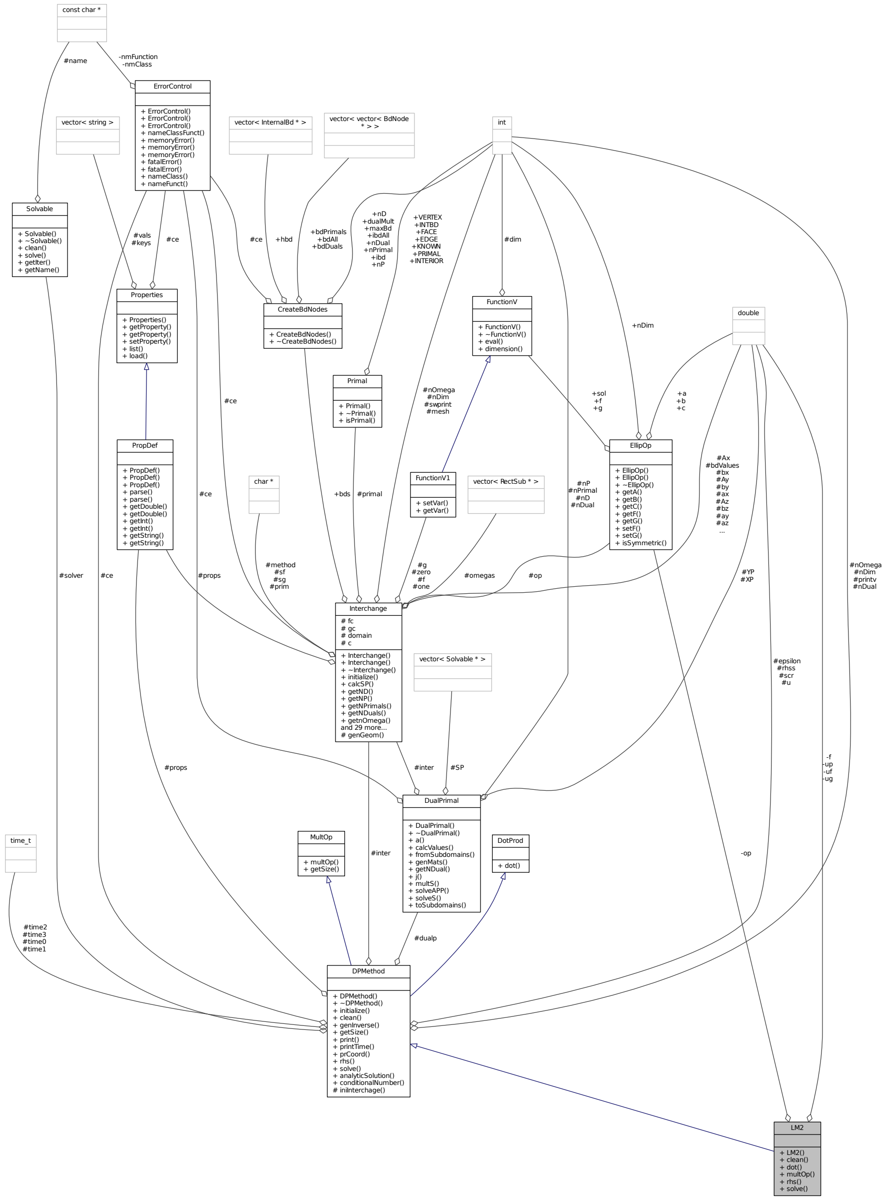
7.36 LM2 Class Reference

```
#include <LM2.hpp>
```

Inheritance diagram for LM2:



Collaboration diagram for LM2:



Public Member Functions

- LM2 (PropDef &props, EllipOp &op)
- virtual void clean (void)
- ldouble dot (ldouble ∗u, ldouble ∗v)
- void multOp (ldouble ∗u, ldouble ∗v)

  y = *A*∗*x*
- void rhs (void)
- void solve (void)

Private Attributes

- ldouble ∗ up
- ldouble ∗ uf
- ldouble ∗ ug
- ldouble ∗ f
- EllipOp ∗ op

Additional Inherited Members

7.36.1 Constructor & Destructor Documentation

7.36.1.1 LM2::LM2 (**PropDef & *props*, EllipOp & *op***) [inline]

7.36.2 Member Function Documentation

7.36.2.1 virtual void LM2::clean (**void**) [inline], [virtual]

Implements DPMethod.

Reimplemented in LM2MPI.

7.36.2.2 ldouble LM2::dot (**ldouble ∗ *u*, ldouble ∗ *v***) [virtual]

Implements DotProd.

7.36.2.3 void LM2::multOp (**ldouble ∗ *x*, ldouble ∗ *y***) [virtual]

y = A∗x

Implements MultOp.

7.36.2.4 void LM2::rhs (**void**) [virtual]

Implements DPMethod.

7.36.2.5 void LM2::solve (**void**) [virtual]

Implements DPMethod.

7.36.3 Member Data Documentation

7.36.3.1 ldouble∗ LM2::f [private]

7.36.3.2 EllipOp∗ LM2::op [private]

7.36.3.3 ldouble∗ LM2::uf [private]

7.36.3.4 ldouble∗ LM2::ug [private]

7.36.3.5 ldouble∗ LM2::up [private]

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

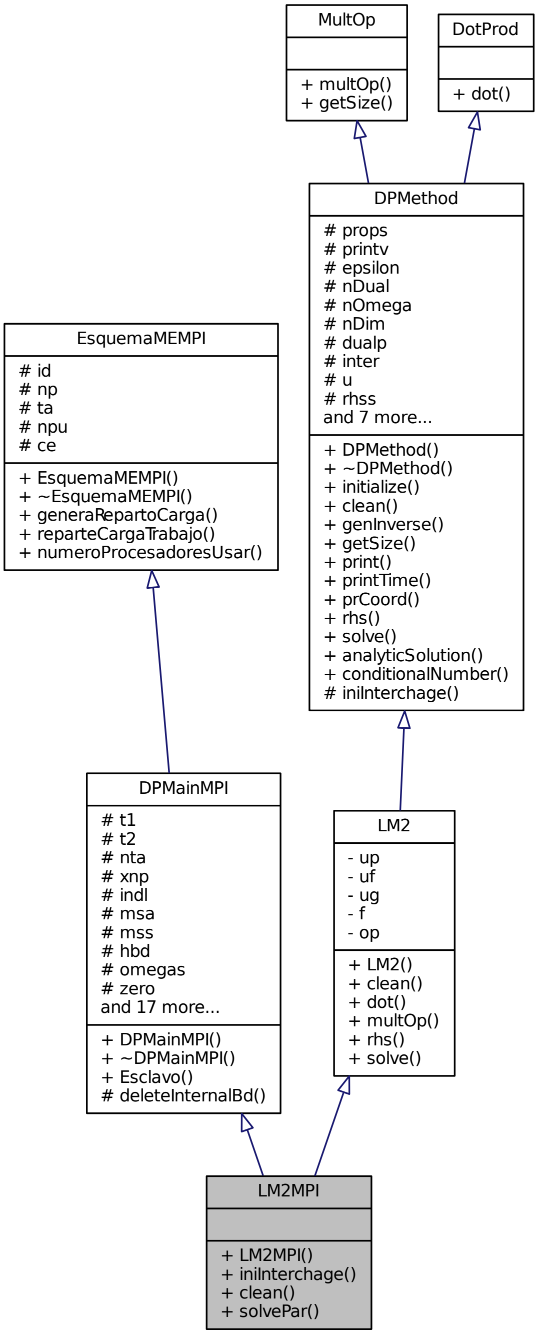
- LM2.hpp
- LM2.cpp

7.37 LM2MPI Class Reference

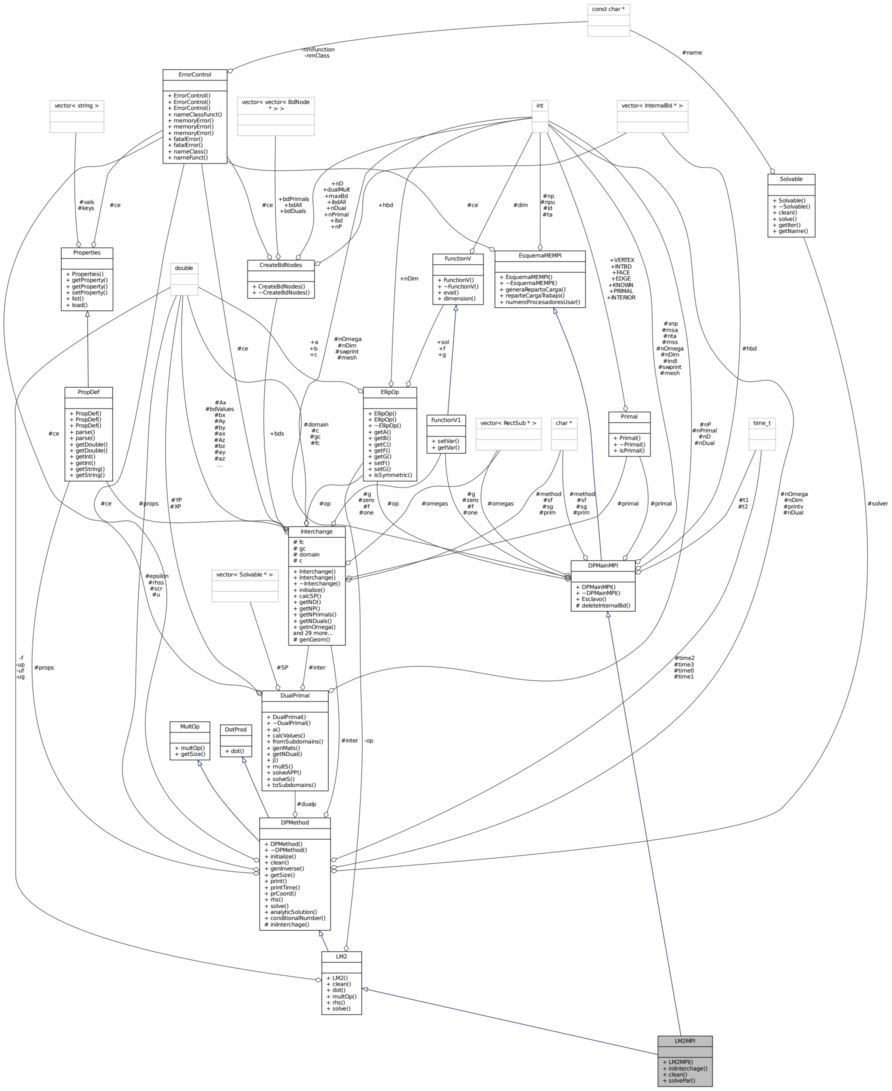
Clase para definir el metodo LM-2 de DVS-DDM.

```
#include <LM2MPI.hpp>
```

Inheritance diagram for LM2MPI:



Collaboration diagram for LM2MPI:



Public Member Functions

- LM2MPI (int id, int np, PropDef &props, EllipOp &op)

  Constructor de la clase.
- void iniInterchage (void)

  Inicializa InterchangeMPI en lugar de Interchange.
- void clean (void)
- void solvePar (void)

  Sobrecarga del la aplicacion.

Additional Inherited Members

7.37.1 Detailed Description

Clase para definir el metodo LM-2 de DVS-DDM.

Clase para definir el metodo LM-2 de DVS-DDM en paralelo

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.37.2 Constructor & Destructor Documentation

7.37.2.1 LM2MPI::LM2MPI (int *id*, int *np*, PropDef & *props*, EllipOp & *op*) [inline]

Constructor de la clase.

7.37.3 Member Function Documentation

7.37.3.1 void LM2MPI::clean (void) [inline], [virtual]

Reimplemented from LM2.

7.37.3.2 void LM2MPI::iniInterchage (void) [inline], [virtual]

Inicializa InterchangeMPI en lugar de Interchange.

Reimplemented from DPMethod.

7.37.3.3 void LM2MPI::solvePar (void) [inline]

Sobrecarga del la aplicacion.

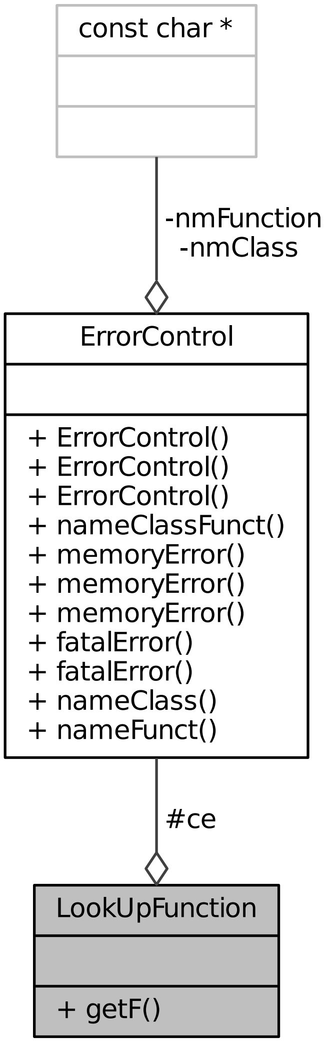
The documentation for this class was generated from the following file:

- LM2MPI.hpp

7.38 LookUpFunction Class Reference

```
#include <LookUpFunction.hpp>
```

Collaboration diagram for LookUpFunction:



Public Member Functions

- FunctionV1 ∗ getF (char ∗s)

Protected Attributes

- ErrorControl ce

  Control de errores.

7.38.1 Member Function Documentation

7.38.1.1 FunctionV1 ∗ LookUpFunction::getF (char ∗ *s*)

7.38.2 Member Data Documentation

7.38.2.1 ErrorControl LookUpFunction::ce `[protected]`

Control de errores.

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

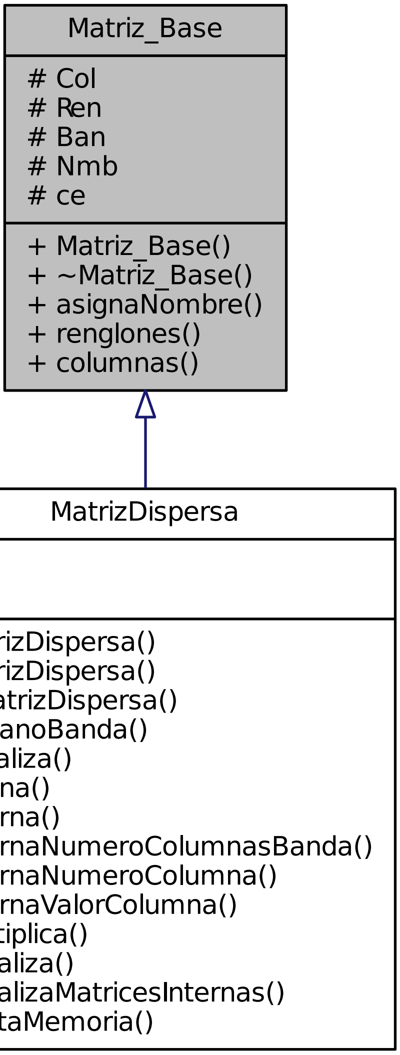
- LookUpFunction.hpp
- LookUpFunction.cpp

7.39 Matriz_Base Class Reference

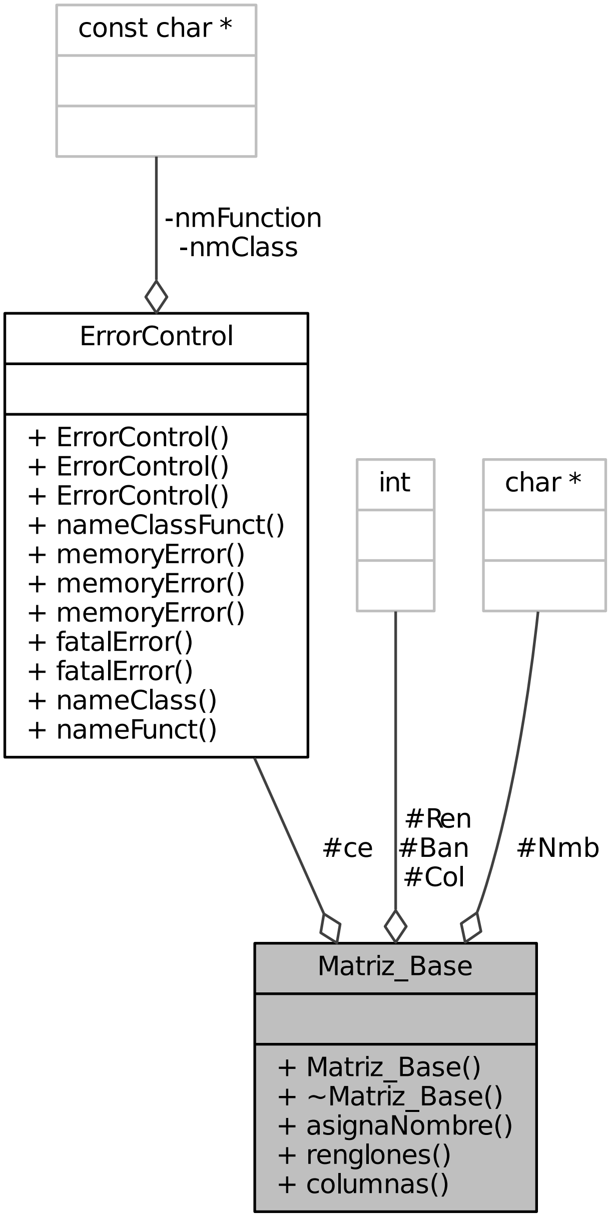
Clase base para el trabajar con matrices.

`#include <Matriz_Base.hpp>`

Inheritance diagram for Matriz_Base:



Collaboration diagram for Matriz_Base:



Public Member Functions

- Matriz_Base (void)

  Constructor de la clase.
- ∼Matriz_Base ()

Destructor de la clase.

- void asignaNombre (const char ∗nmb)

  Asigna nombre a la matriz.
- int renglones (void)

  Retorna el numero de renglones de la matriz.
- int columnas (void)

  Retorna el numero de columnas de la matriz.

Protected Attributes

- int Col

  Numero de columnas.
- int Ren

  Numero de renglones.
- int Ban

  Tamano de la banda (solo si es bandada o dispersa)
- char ∗ Nmb

  Nombre de la matriz.
- ErrorControl ce

  Control de errores.

7.39.1 Detailed Description

Clase base para el trabajar con matrices.

Author

Antonio Carrillo Ledesma

Date

primavera 2009

Version

1.0.1

Bug No hay errores conocidos

7.39.2 Constructor & Destructor Documentation

7.39.2.1 Matriz_Base::Matriz_Base (void) [inline]

Constructor de la clase.

7.39.2.2 Matriz_Base::∼Matriz_Base ( ) [inline]

Destructor de la clase.

7.39.3 Member Function Documentation

7.39.3.1 void Matriz_Base::asignaNombre (const char ∗ *nmb*) [inline]

Asigna nombre a la matriz.

Parameters

nmb	Nombre de la matriz

7.39.3.2 int Matriz_Base::columnas (void) [inline]

Retorna el numero de columnas de la matriz.

Returns

Regresa el numero de columnas de la matriz

7.39.3.3 int Matriz_Base::renglones (void) [inline]

Retorna el numero de renglones de la matriz.

Returns

Regresa el numero de renglones de la matriz

7.39.4 Member Data Documentation

7.39.4.1 int Matriz_Base::Ban [protected]

Tamano de la banda (solo si es bandada o dispersa)

7.39.4.2 ErrorControl Matriz_Base::ce [protected]

Control de errores.

7.39.4.3 int Matriz_Base::Col [protected]

Numero de columnas.

7.39.4.4 char∗ Matriz_Base::Nmb [protected]

Nombre de la matriz.

7.39.4.5 int Matriz_Base::Ren [protected]

Numero de renglones.

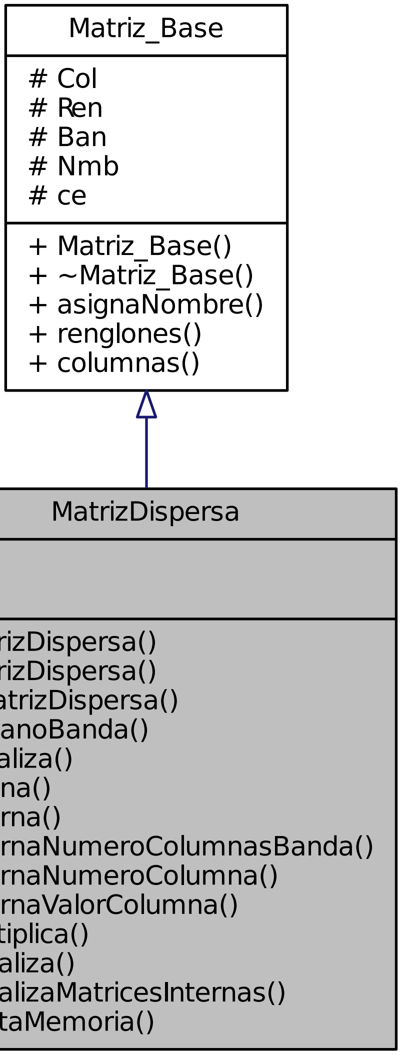
The documentation for this class was generated from the following file:

- Matriz_Base.hpp

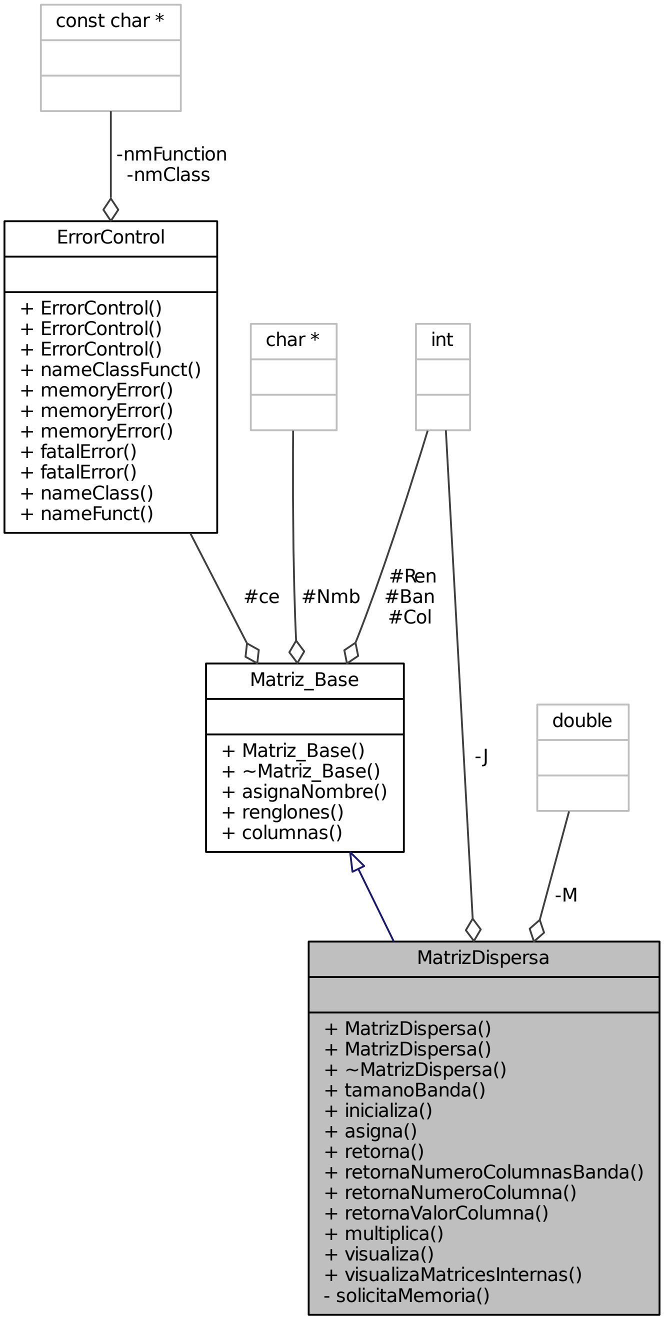
7.40 MatrizDispersa Class Reference

```
#include <MatrizDispersa.hpp>
```

Inheritance diagram for MatrizDispersa:



Collaboration diagram for MatrizDispersa:



Public Member Functions

- MatrizDispersa (const int ren, const int col, const int ban)

Constructor de la clase.

- MatrizDispersa (const int ren, const int col, const int ban, const char ∗nmb)

  Constructor de la clase.
- ∼MatrizDispersa ()
- int tamanoBanda (void)

  Retorna el tamano de la banda.
- void inicializa (ldouble val)

  Inicializa la matriz al valor indicado.
- void asigna (const int ren, const int col, const ldouble val)

  Asigna el valor indicado en el renglo y columna solicitado.
- ldouble retorna (const int ren, const int col)

  Retorna el numero de columna cuando se para en el renglon e indice de la banda.
- int retornaNumeroColumnasBanda (int ren)

  Retorna el numero de columnas de la banda para el renglon indicado.
- int retornaNumeroColumna (int ren, int ind)

  Retorna el numero de columna cuando se para en el renglon e indice de la banda.
- ldouble retornaValorColumna (int ren, int ind)

  Retorna el valor de la columna cuando se para en el renglon e indice de la banda.
- void multiplica (ldouble ∗b, ldouble ∗r)

  Multiplica la matriz por el vector B dejando el Resultado en R.
- void visualiza (const int tp)

  Visualiza la matriz.
- void visualizaMatricesInternas (void)

  Visualiza las matrices internas usadas para soportar a las matrices bandadas.

Private Member Functions

- void solicitaMemoria (const int ren, const int col, const int ban)

  Solicita la memoria necesaria para contener los valores de la matriz.

Private Attributes

- ldouble ∗∗ M

  Puntero a la matriz de datos.
- int ∗∗ J

  Arreglo que contiene los columnas de la matriz.

Additional Inherited Members

7.40.1 Detailed Description

Clase para el trabajar con matrices dispersas de punto flotante basada en el algoritmo Jagged Diagonal Storage (JDS) El algoritmo esta optimizado para hacer producto matriz vector

Author

Antonio Carrillo Ledesma

Date

primavera 2009

Version

1.0.1

Bug No hay errores conocidos

Todo Hacer comportamiento para cambiar tamano de banda

Multiplicacion de matrices

7.40.2 Constructor & Destructor Documentation

7.40.2.1 MatrizDispersa::MatrizDispersa (const int *ren*, const int *col*, const int *ban*) [inline]

Constructor de la clase.

Parameters

ren	Numero de renglones de la matriz
col	Numero de columnas de la matriz
ban	Tamano de la banda

7.40.2.2 MatrizDispersa::MatrizDispersa (const int *ren*, const int *col*, const int *ban*, const char ∗ *nmb*) [inline]

Constructor de la clase.

Parameters

ren	Numero de renglones de la matriz
col	Numero de columnas de la matriz
ban	Tamano de la banda
nmb	Nombre de la matriz

7.40.2.3 MatrizDispersa::∼MatrizDispersa ( ) [inline]

7.40.3 Member Function Documentation

7.40.3.1 void MatrizDispersa::asigna (const int *ren*, const int *col*, const ldouble *val*)

Asigna el valor indicado en el renglo y columna solicitado.

Parameters

ren	Renglon
col	Columna

val	Valor

7.40.3.2 void MatrizDispersa::inicializa (ldouble *val*) [inline]

Inicializa la matriz al valor indicado.

Parameters

val	Valor por omision para inicializar la matriz

7.40.3.3 void MatrizDispersa::multiplica (ldouble ∗ *b,* ldouble ∗ *r*)

Multiplica la matriz por el vector B dejando el Resultado en R.

Parameters

b	Puntero a un Vector
r	Puntero a un Vector

7.40.3.4 ldouble MatrizDispersa::retorna (const int *ren,* const int *col*)

Retorna el numero de columna cuando se para en el renglon e indice de la banda.

Parameters

ren	Numero de renglon
col	Numero de columna

Returns

Numero de columna cuando se para en el renglon e indice de la banda

7.40.3.5 int MatrizDispersa::retornaNumeroColumna (int *ren,* int *ind*) [inline]

Retorna el numero de columna cuando se para en el renglon e indice de la banda.

Parameters

ren	Numero de renglon
ind	Numero de indice

Returns

Numero de columna cuando se para en el renglon e indice de la banda

7.40.3.6 int MatrizDispersa::retornaNumeroColumnasBanda (int *ren*)

Retorna el numero de columnas de la banda para el renglon indicado.

Parameters

ren	Numero de renglon

Returns

Numero de columnas de la banda para el renglon solicitado

7.40.3.7 ldouble MatrizDispersa::retornaValorColumna (int *ren,* int *ind*) [inline]

Retorna el valor de la columna cuando se para en el renglon e indice de la banda.

Parameters

ren	Numero de renglon
ind	Numero de indice

Returns

Valor de la columna cuando se para en el renglon e indice de la banda

7.40.3.8 void MatrizDispersa::solicitaMemoria (const int *ren,* const int *col,* const int *ban*) [private]

Solicita la memoria necesaria para contener los valores de la matriz.

Parameters

ren	Numero de renglones de la matriz
col	Numero de columnas de la matriz
ban	Tamano de la banda

7.40.3.9 int MatrizDispersa::tamanoBanda (void) [inline]

Retorna el tamano de la banda.

Returns

Tamano de la banda

7.40.3.10 void MatrizDispersa::visualiza (const int *tp*)

Visualiza la matriz.

Parameters

tp	(1) Se visualiza el vector de en formato de notacion cientifica, (0) formato notacion de punto flotante

7.40.3.11 void MatrizDispersa::visualizaMatricesInternas (void)

Visualiza las matrices internas usadas para soportar a las matrices bandadas.

7.40.4 Member Data Documentation

7.40.4.1 int∗∗ MatrizDispersa::J [private]

Arreglo que contiene los columnas de la matriz.

7.40.4.2 ldouble∗∗ MatrizDispersa::M [private]

Puntero a la matriz de datos.

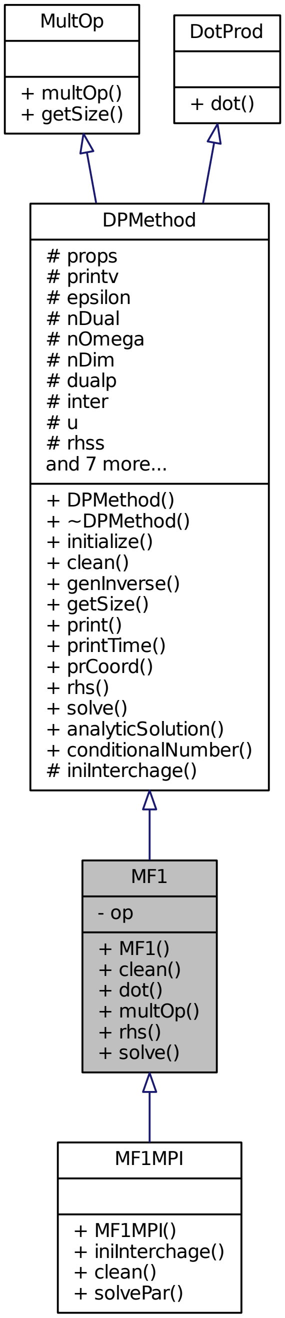
The documentation for this class was generated from the following files:

- MatrizDispersa.hpp
- MatrizDispersa.cpp

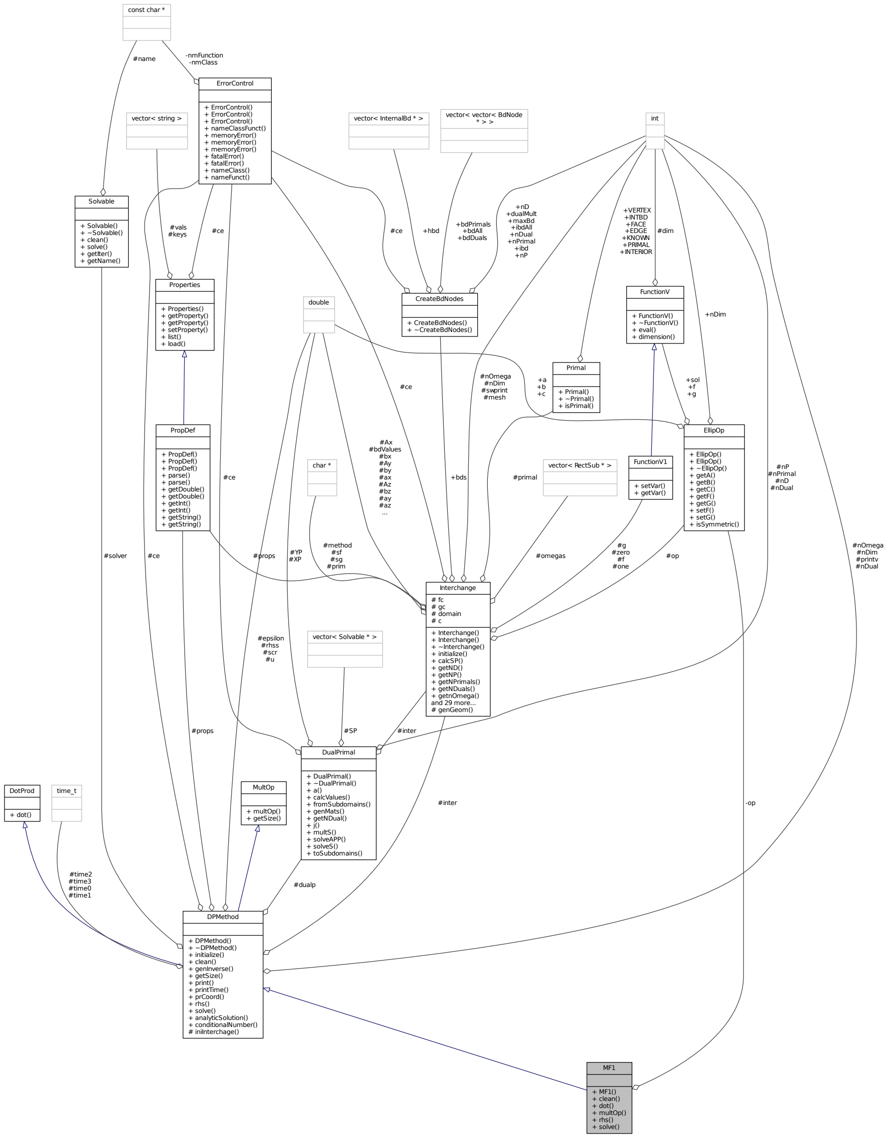
7.41 MF1 Class Reference

```
#include <MF1.hpp>
```

Inheritance diagram for MF1:



Collaboration diagram for MF1:



Public Member Functions

- MF1 (PropDef &props, EllipOp &op)
- virtual void clean (void)
- ldouble dot (ldouble ∗u, ldouble ∗v)
- void multOp (ldouble ∗u, ldouble ∗v)

  $y = A*x$
- void rhs (void)
- void solve (void)

Private Attributes

- EllipOp ∗ op

Additional Inherited Members

7.41.1 Constructor & Destructor Documentation

7.41.1.1 MF1::MF1 (PropDef & *props*, EllipOp & *op*) [inline]

7.41.2 Member Function Documentation

7.41.2.1 virtual void MF1::clean (void) [inline], [virtual]

Implements DPMethod.

Reimplemented in MF1MPI.

7.41.2.2 ldouble MF1::dot (ldouble ∗ *u*, ldouble ∗ *v*) [virtual]

Implements DotProd.

7.41.2.3 void MF1::multOp (ldouble ∗ *x*, ldouble ∗ *y*) [virtual]

y = A∗x

Implements MultOp.

7.41.2.4 void MF1::rhs (void) [virtual]

Implements DPMethod.

7.41.2.5 void MF1::solve (void) [virtual]

Implements DPMethod.

7.41.3 Member Data Documentation

7.41.3.1 EllipOp∗ MF1::op [private]

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

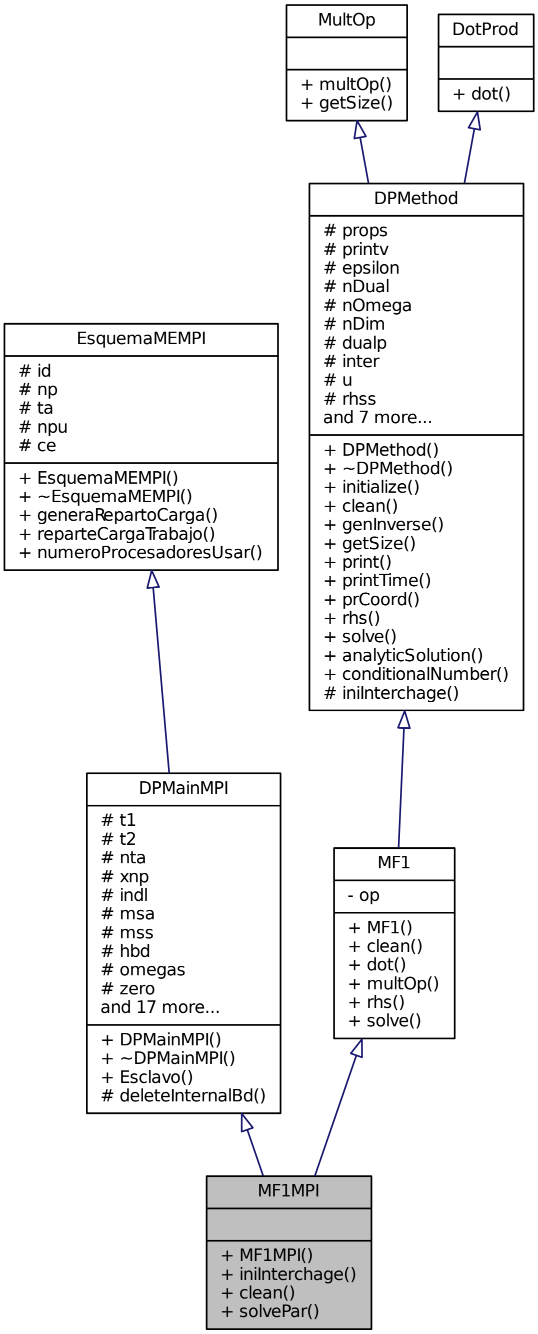
- MF1.hpp
- MF1.cpp

7.42 MF1MPI Class Reference

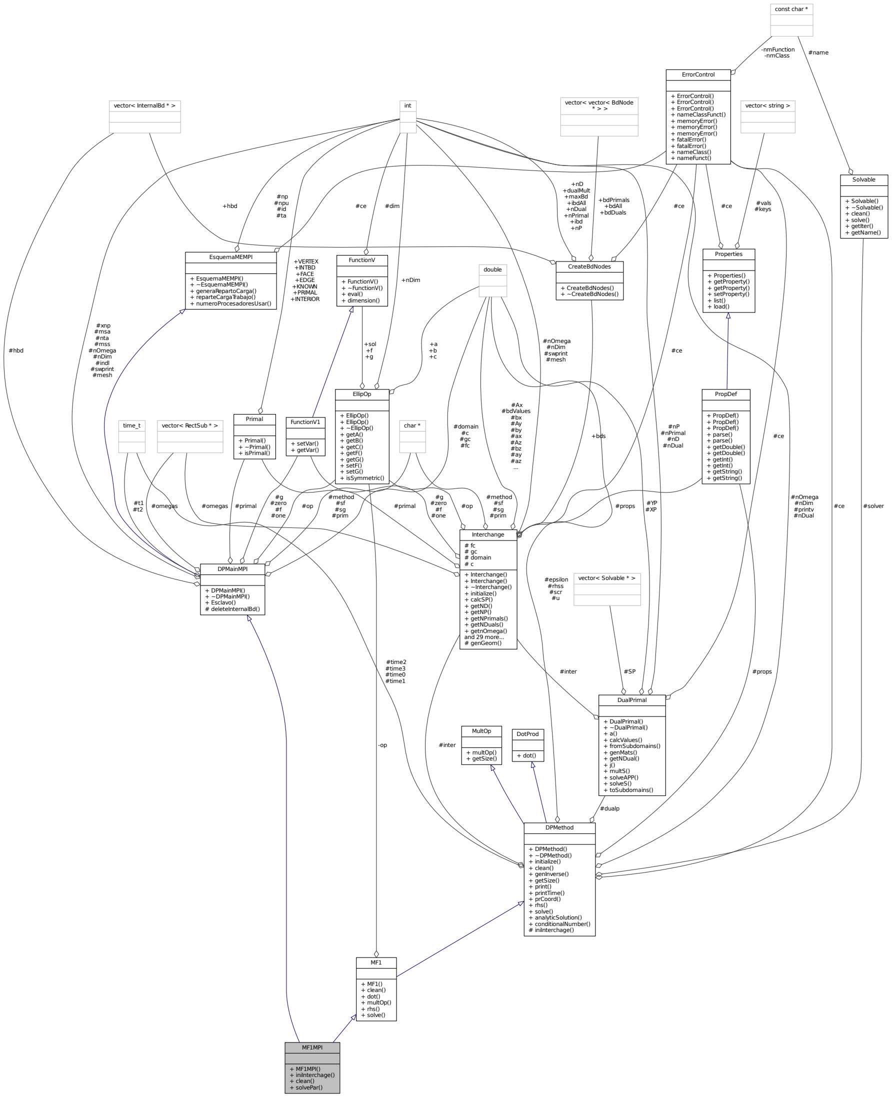
Clase para definir el metodo MF-1 de DVS-DDM.

```
#include <MF1MPI.hpp>
```

Inheritance diagram for MF1MPI:



Collaboration diagram for MF1MPI:



Public Member Functions

- MF1MPI (int id, int np, PropDef &props, EllipOp &op)

    Constructor de la clase.
- void iniInterchage (void)

    Inicializa InterchangeMPI en lugar de Interchange.
- void clean (void)
- void solvePar (void)

    Sobrecarga del la aplicacion.

Additional Inherited Members

7.42.1 Detailed Description

Clase para definir el metodo MF-1 de DVS-DDM.

Clase para definir el metodo MF-1 de DVS-DDM en paralelo

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.42.2 Constructor & Destructor Documentation

7.42.2.1 MF1MPI::MF1MPI (int *id*, int *np*, PropDef & *props*, EllipOp & *op*) [inline]

Constructor de la clase.

7.42.3 Member Function Documentation

7.42.3.1 void MF1MPI::clean (void) [inline], [virtual]

Reimplemented from MF1.

7.42.3.2 void MF1MPI::iniInterchage (void) [inline], [virtual]

Inicializa InterchangeMPI en lugar de Interchange.

Reimplemented from DPMethod.

7.42.3.3 void MF1MPI::solvePar (void) [inline]

Sobrecarga del la aplicacion.

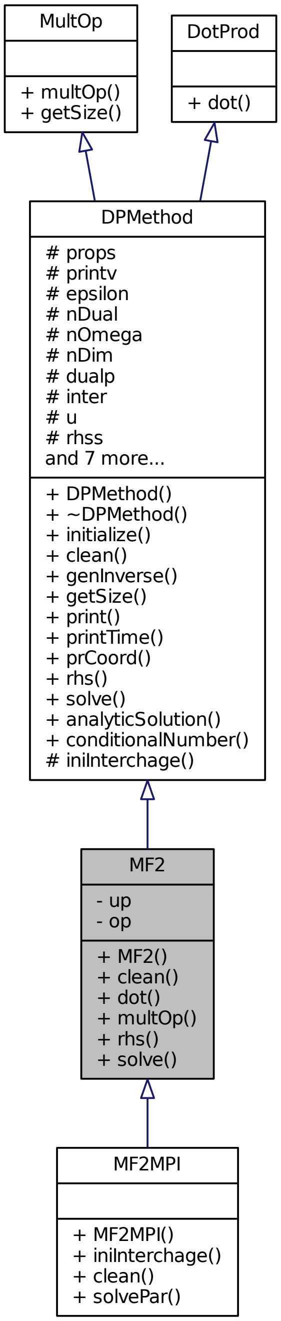
The documentation for this class was generated from the following file:

- MF1MPI.hpp

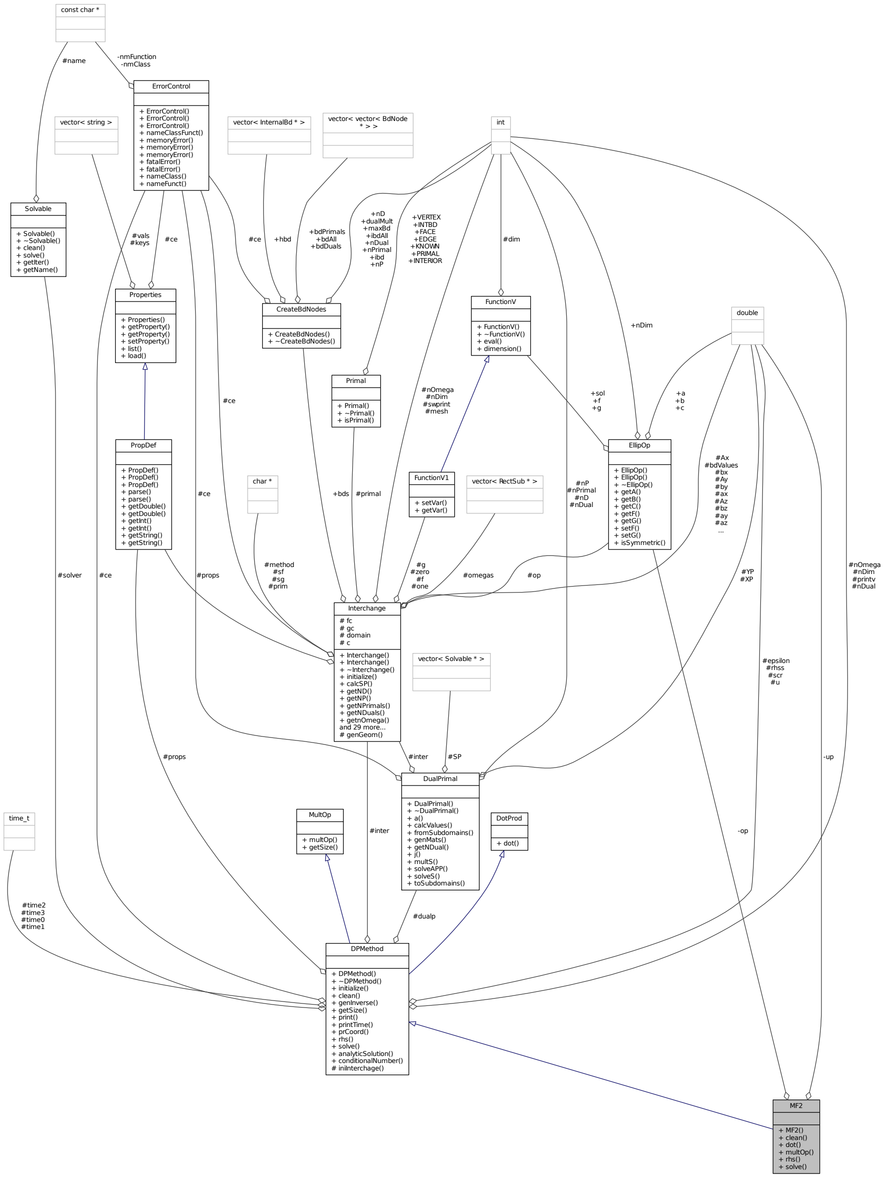
7.43 MF2 Class Reference

```
#include <MF2.hpp>
```

Inheritance diagram for MF2:



Collaboration diagram for MF2:



Public Member Functions

- MF2 (PropDef &props, EllipOp &op)
- virtual void clean (void)
- ldouble dot (ldouble ∗u, ldouble ∗v)
- void multOp (ldouble ∗u, ldouble ∗v)

  $y = A*x$
- void rhs (void)
- void solve (void)

Private Attributes

- ldouble ∗ up
- EllipOp ∗ op

Additional Inherited Members

7.43.1 Constructor & Destructor Documentation

7.43.1.1 MF2::MF2 (**PropDef** & *props*, **EllipOp** & *op*) [inline]

7.43.2 Member Function Documentation

7.43.2.1 virtual void MF2::clean (void) [inline], [virtual]

Implements DPMethod.

Reimplemented in MF2MPI.

7.43.2.2 ldouble MF2::dot (**ldouble** ∗ *u*, **ldouble** ∗ *v*) [virtual]

Implements DotProd.

7.43.2.3 void MF2::multOp (**ldouble** ∗ *x*, **ldouble** ∗ *y*) [virtual]

y = A∗x

Implements MultOp.

7.43.2.4 void MF2::rhs (void) [virtual]

Implements DPMethod.

7.43.2.5 void MF2::solve (void) [virtual]

Implements DPMethod.

7.43.3 Member Data Documentation

7.43.3.1 EllipOp∗ MF2::op [private]

7.43.3.2 ldouble∗ MF2::up [private]

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

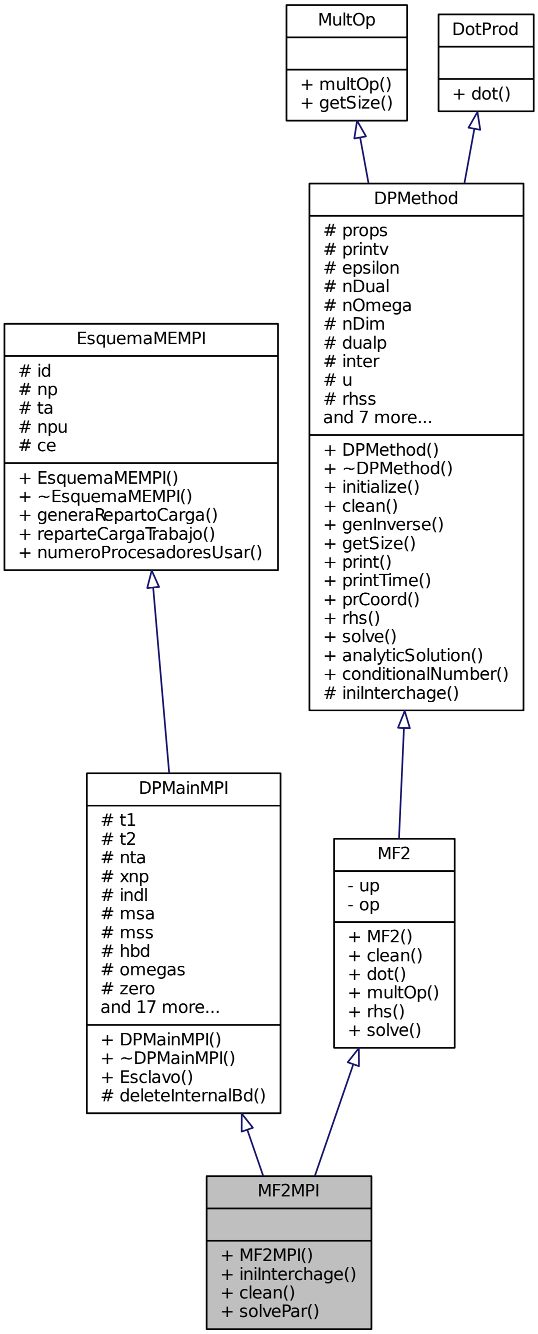
- MF2.hpp
- MF2.cpp

7.44 MF2MPI Class Reference

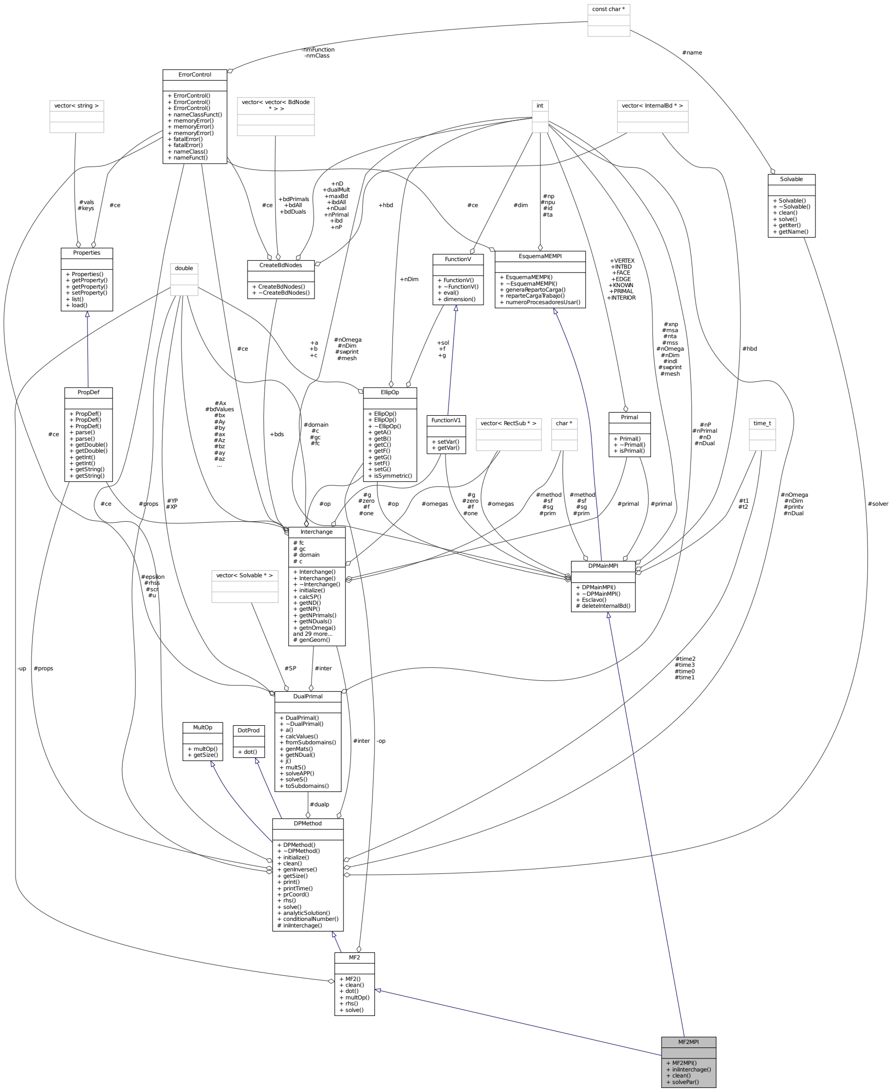
Clase para definir el metodo MF-2 de DVS-DDM.

```
#include <MF2MPI.hpp>
```

Inheritance diagram for MF2MPI:



Collaboration diagram for MF2MPI:



Public Member Functions

- MF2MPI (int id, int np, PropDef &props, EllipOp &op)

  Constructor de la clase.
- void iniInterchage (void)

  Inicializa InterchangeMPI en lugar de Interchange.
- void clean (void)
- void solvePar (void)

  Sobrecarga del la aplicacion.

Additional Inherited Members

7.44.1 Detailed Description

Clase para definir el metodo MF-2 de DVS-DDM.

Clase para definir el metodo MF-2 de DVS-DDM en paralelo

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.44.2 Constructor & Destructor Documentation

7.44.2.1 MF2MPI::MF2MPI (int *id*, int *np*, PropDef & *props*, EllipOp & *op*) [inline]

Constructor de la clase.

7.44.3 Member Function Documentation

7.44.3.1 void MF2MPI::clean (void) [inline], [virtual]

Reimplemented from MF2.

7.44.3.2 void MF2MPI::iniInterchage (void) [inline], [virtual]

Inicializa InterchangeMPI en lugar de Interchange.

Reimplemented from DPMethod.

7.44.3.3 void MF2MPI::solvePar (void) [inline]

Sobrecarga del la aplicacion.

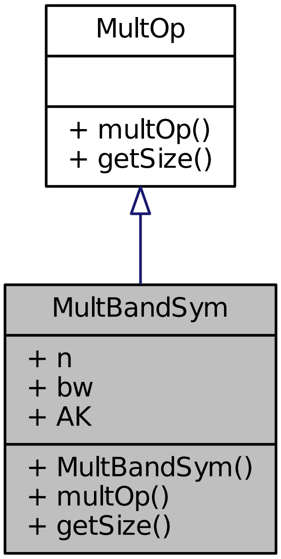
The documentation for this class was generated from the following file:

- MF2MPI.hpp

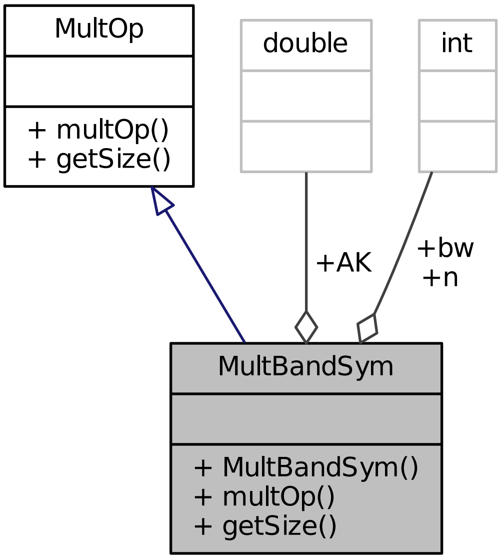
7.45 MultBandSym Class Reference

```
#include <MultBandSym.hpp>
```

Inheritance diagram for MultBandSym:



Collaboration diagram for MultBandSym:



Public Member Functions

- MultBandSym (int n, int bw, ldouble ∗∗AK)
- void multOp (ldouble ∗x, ldouble ∗y)

  y = A∗x
- int getSize (void)

  vector size

Public Attributes

- int n
- int bw
- ldouble ∗∗ AK

7.45.1 Constructor & Destructor Documentation

7.45.1.1 MultBandSym::MultBandSym (int *n*, int *bw*, ldouble ∗∗ *AK*) [inline]

7.45.2 Member Function Documentation

7.45.2.1 int MultBandSym::getSize (void) [inline], [virtual]

vector size

Implements MultOp.

7.45.2.2 void MultBandSym::multOp (Idouble ∗ *x*, Idouble ∗ *y*) [virtual]

y = A∗x

Implements MultOp.

7.45.3 Member Data Documentation

7.45.3.1 Idouble∗∗ MultBandSym::AK

7.45.3.2 int MultBandSym::bw

7.45.3.3 int MultBandSym::n

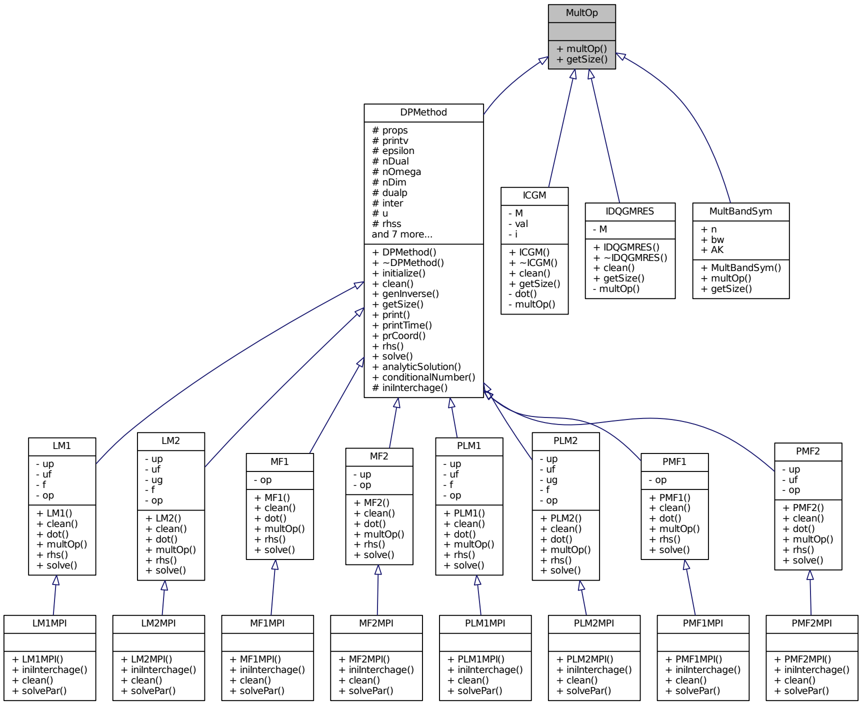
The documentation for this class was generated from the following files:

- MultBandSym.hpp
- MultBandSym.cpp

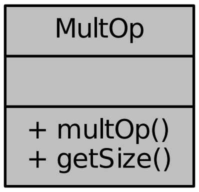
7.46 MultOp Class Reference

```
#include <MultOp.hpp>
```

Inheritance diagram for MultOp:



Collaboration diagram for MultOp:



Public Member Functions

- virtual void multOp (ldouble ∗x, ldouble ∗y)=0

  y = A∗x
- virtual int getSize (void)=0

  vector size

7.46.1 Member Function Documentation

7.46.1.1 virtual int MultOp::getSize (void) `[pure virtual]`

vector size

Implemented in DPMethod, ICGM, IDQGMRES, and MultBandSym.

7.46.1.2 virtual void MultOp::multOp (ldouble ∗ *x*, ldouble ∗ *y*) `[pure virtual]`

y = A∗x

Implemented in LM2, PLM1, PLM2, LM1, MF2, PMF2, ICGM, PMF1, MF1, IDQGMRES, and MultBandSym.

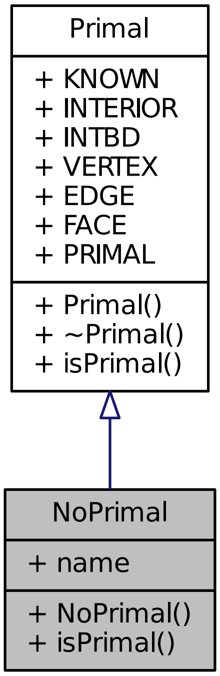
The documentation for this class was generated from the following file:

- MultOp.hpp

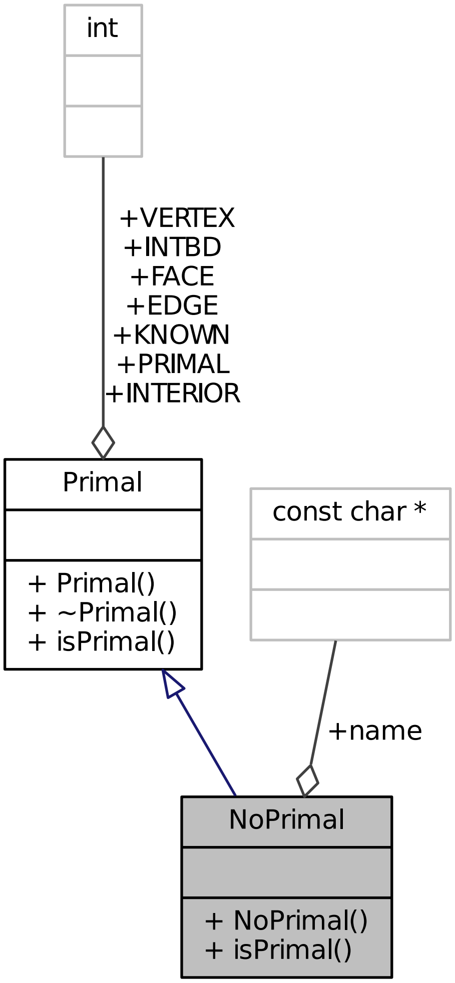
7.47 NoPrimal Class Reference

```
#include <NoPrimal.hpp>
```

Inheritance diagram for NoPrimal:



Collaboration diagram for NoPrimal:



Public Member Functions

- NoPrimal (void)
- bool isPrimal (int type, int ∗coordN, int ∗coordM)

Public Attributes

- const char ∗ name

Additional Inherited Members

7.47.1 Constructor & Destructor Documentation

7.47.1.1 NoPrimal::NoPrimal (void) [inline]

7.47.2 Member Function Documentation

7.47.2.1 bool NoPrimal::isPrimal (int *type*, int ∗ *coordN*, int ∗ *coordM*) [inline], [virtual]

Implements Primal.

7.47.3 Member Data Documentation

7.47.3.1 const char∗ NoPrimal::name

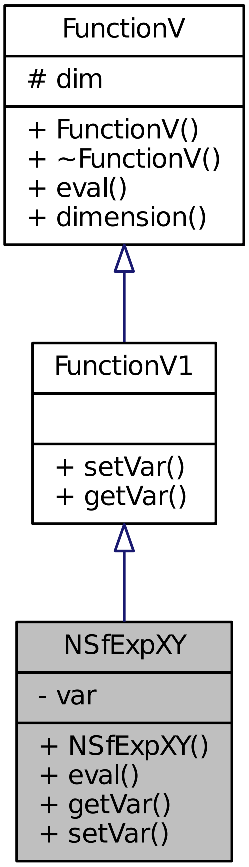
The documentation for this class was generated from the following file:

- NoPrimal.hpp

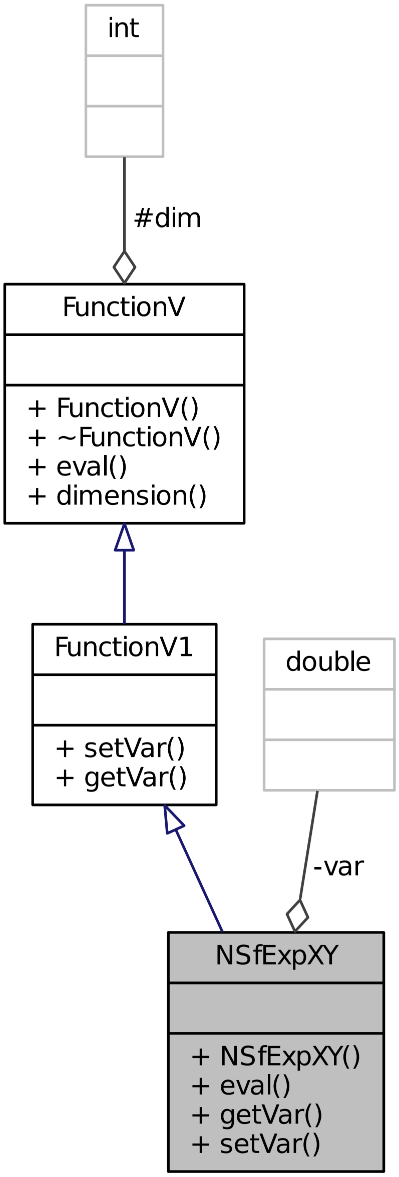
7.48 NSfExpXY Class Reference

```
#include <NSfExpXY.hpp>
```

Inheritance diagram for NSfExpXY:



Collaboration diagram for NSfExpXY:



Public Member Functions

- NSfExpXY (double b)
- double eval (int d, double ∗x)
- double getVar (void)
- void setVar (double b)

Private Attributes

- double var

Additional Inherited Members

7.48.1 Constructor & Destructor Documentation

7.48.1.1 NSfExpXY::NSfExpXY (double *b*) [inline]

7.48.2 Member Function Documentation

7.48.2.1 double NSfExpXY::eval (int *d*, double ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.48.2.2 double NSfExpXY::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.48.2.3 void NSfExpXY::setVar (double *b*) [inline], [virtual]

Implements FunctionV1.

7.48.3 Member Data Documentation

7.48.3.1 double NSfExpXY::var [private]

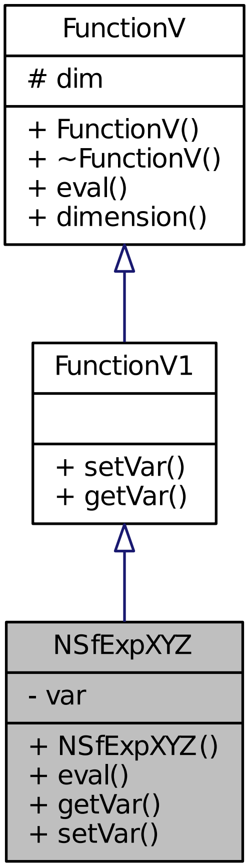
The documentation for this class was generated from the following file:

- NSfExpXY.hpp

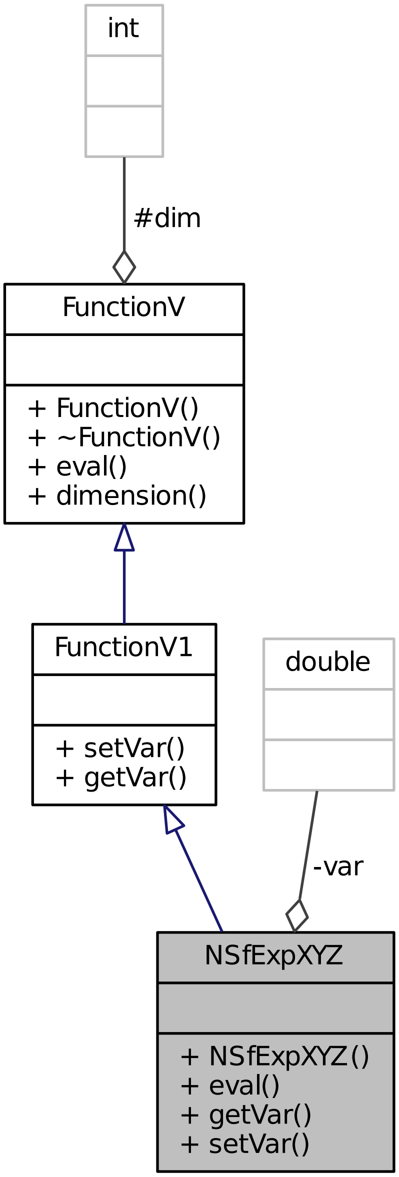
7.49 NSfExpXYZ Class Reference

```
#include <NSfExpXYZ.hpp>
```

Inheritance diagram for NSfExpXYZ:



Collaboration diagram for NSfExpXYZ:



Public Member Functions

- NSfExpXYZ (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.49.1 Constructor & Destructor Documentation

7.49.1.1 NSfExpXYZ::NSfExpXYZ (Idouble *b*) [inline]

7.49.2 Member Function Documentation

7.49.2.1 Idouble NSfExpXYZ::eval (int *d*, Idouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.49.2.2 Idouble NSfExpXYZ::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.49.2.3 void NSfExpXYZ::setVar (Idouble *b*) [inline], [virtual]

Implements FunctionV1.

7.49.3 Member Data Documentation

7.49.3.1 Idouble NSfExpXYZ::var [private]

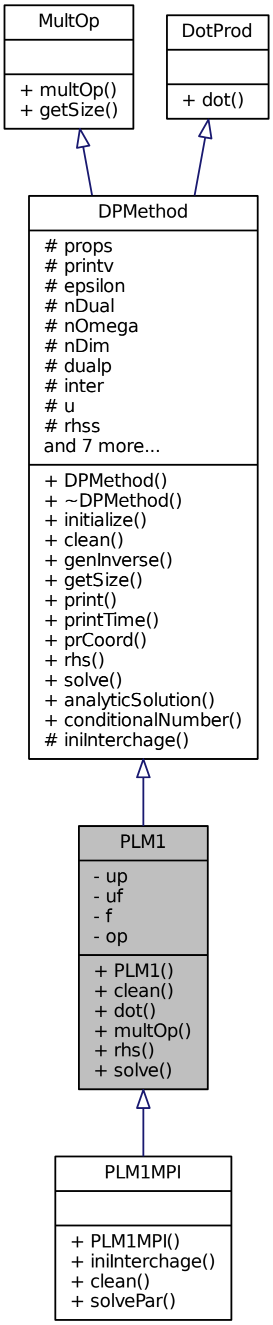
The documentation for this class was generated from the following file:

- NSfExpXYZ.hpp

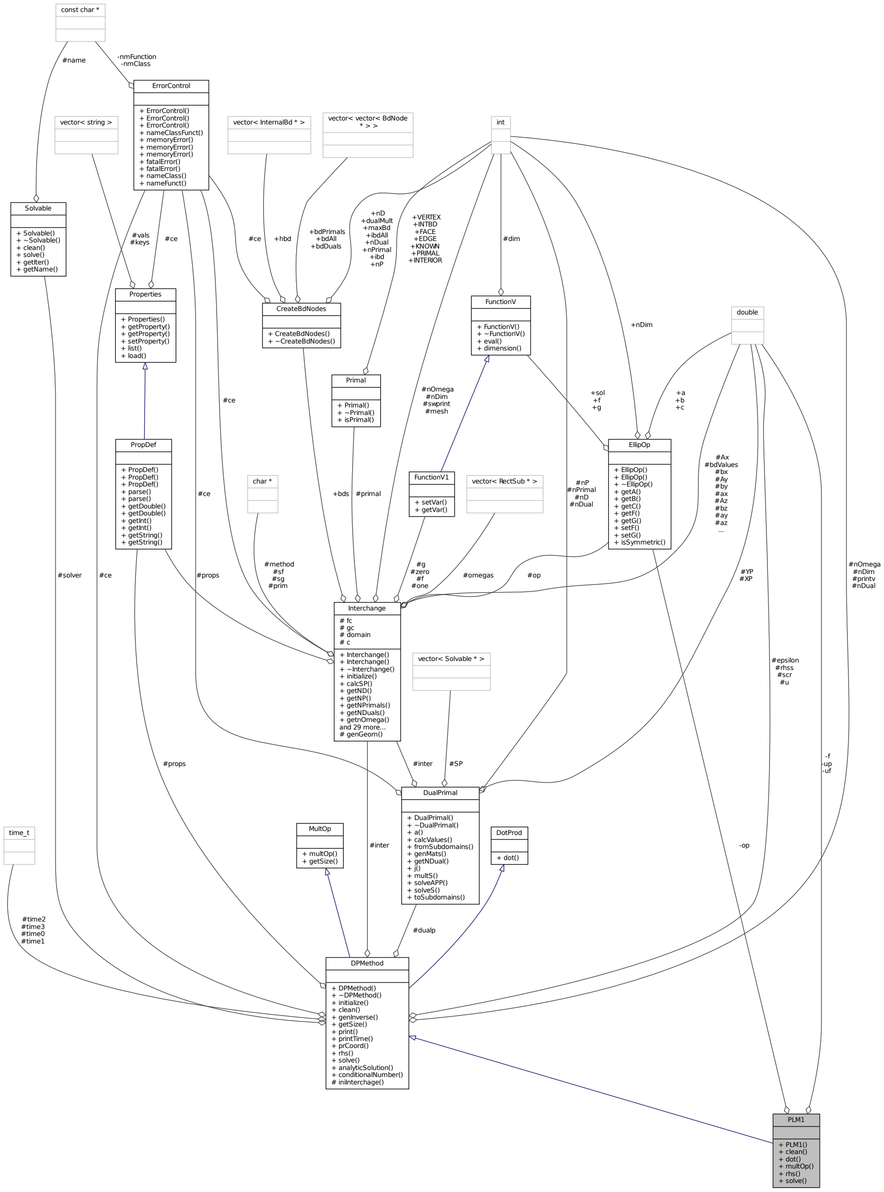
7.50 PLM1 Class Reference

```
#include <PLM1.hpp>
```

Inheritance diagram for PLM1:



Collaboration diagram for PLM1:



Public Member Functions

- PLM1 (PropDef &props, EllipOp &op)
- virtual void clean (void)
- ldouble dot (ldouble $*$u, ldouble $*$v)
- void multOp (ldouble $*$u, ldouble $*$v)

  $y = A*x$
- void rhs (void)
- void solve (void)

Private Attributes

- ldouble ∗ up
- ldouble ∗ uf
- ldouble ∗ f
- EllipOp ∗ op

Additional Inherited Members

7.50.1 Constructor & Destructor Documentation

7.50.1.1 PLM1::PLM1 (PropDef & *props*, EllipOp & *op*) [inline]

7.50.2 Member Function Documentation

7.50.2.1 virtual void PLM1::clean (void) [inline], [virtual]

Implements DPMethod.

Reimplemented in PLM1MPI.

7.50.2.2 ldouble PLM1::dot (ldouble ∗ *u*, ldouble ∗ *v*) [virtual]

Implements DotProd.

7.50.2.3 void PLM1::multOp (ldouble ∗ *x*, ldouble ∗ *y*) [virtual]

y = A∗x

Implements MultOp.

7.50.2.4 void PLM1::rhs (void) [virtual]

Implements DPMethod.

7.50.2.5 void PLM1::solve (void) [virtual]

Implements DPMethod.

7.50.3 Member Data Documentation

7.50.3.1 ldouble∗ PLM1::f [private]

7.50.3.2 EllipOp∗ PLM1::op [private]

7.50.3.3 ldouble∗ PLM1::uf [private]

7.50.3.4 ldouble∗ PLM1::up [private]

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

- PLM1.hpp

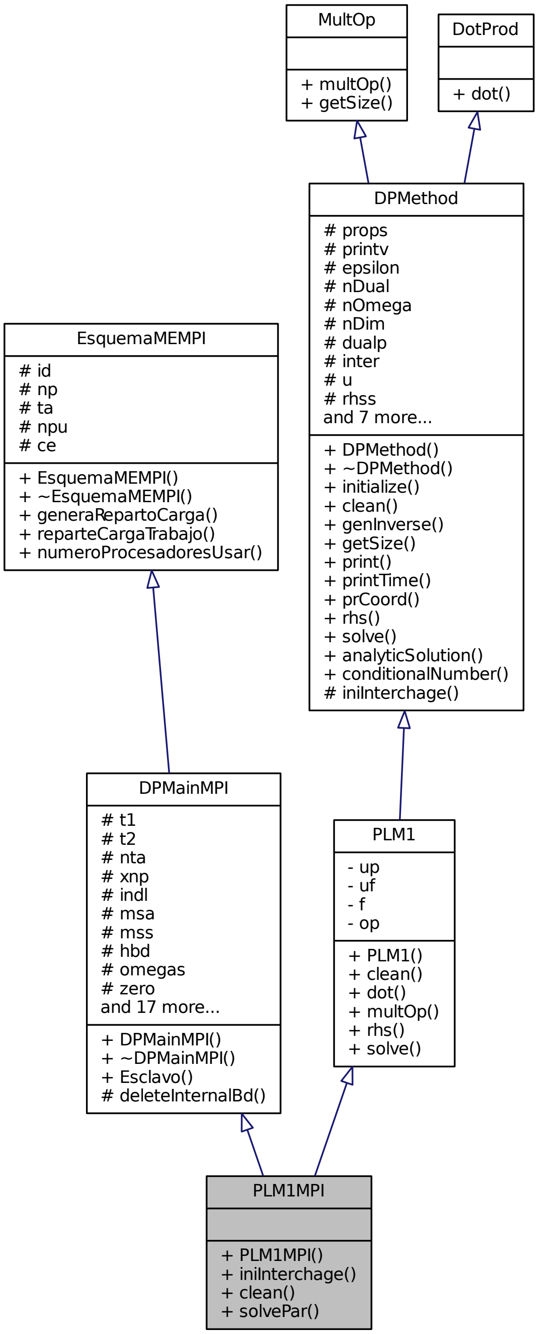
- PLM1.cpp

7.51 PLM1MPI Class Reference

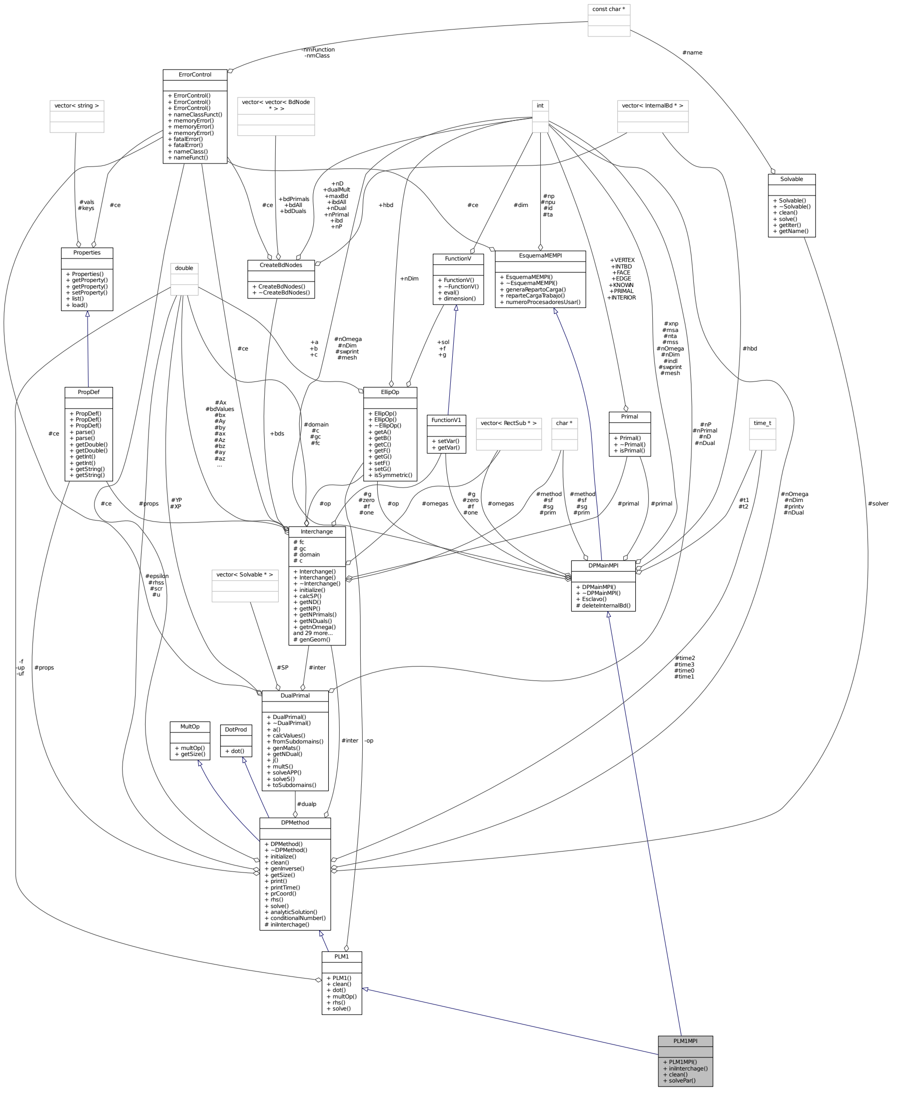
Clase para definir el metodo PLM-1 de DVS-DDM.

```
#include <PLM1MPI.hpp>
```

Inheritance diagram for PLM1MPI:



Collaboration diagram for PLM1MPI:



Public Member Functions

- PLM1MPI (int id, int np, PropDef &props, EllipOp &op)

  Constructor de la clase.
- void iniInterchage (void)

  Inicializa InterchangeMPI en lugar de Interchange.
- void clean (void)
- void solvePar (void)

  Sobrecarga del la aplicacion.

Additional Inherited Members

7.51.1 Detailed Description

Clase para definir el metodo PLM-1 de DVS-DDM.

Clase para definir el metodo PLM-1 de DVS-DDM en paralelo

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.51.2 Constructor & Destructor Documentation

7.51.2.1 PLM1MPI::PLM1MPI (int *id*, int *np*, PropDef & *props*, EllipOp & *op*) [inline]

Constructor de la clase.

7.51.3 Member Function Documentation

7.51.3.1 void PLM1MPI::clean (void) [inline], [virtual]

Reimplemented from PLM1.

7.51.3.2 void PLM1MPI::iniInterchage (void) [inline], [virtual]

Inicializa InterchangeMPI en lugar de Interchange.

Reimplemented from DPMethod.

7.51.3.3 void PLM1MPI::solvePar (void) [inline]

Sobrecarga del la aplicacion.

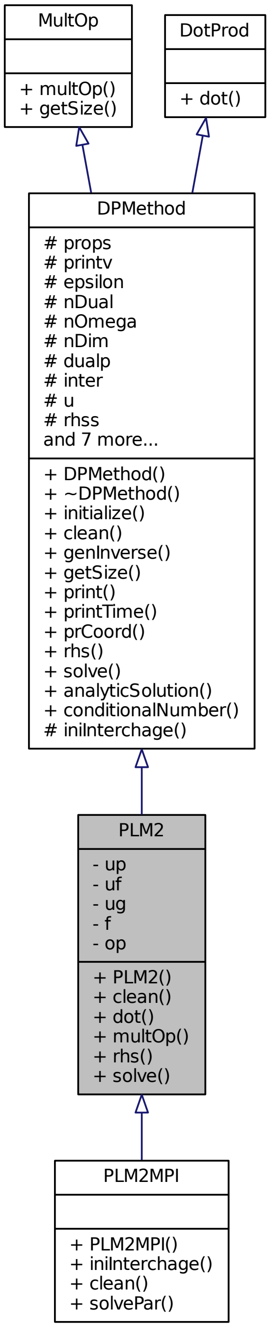
The documentation for this class was generated from the following file:

- PLM1MPI.hpp

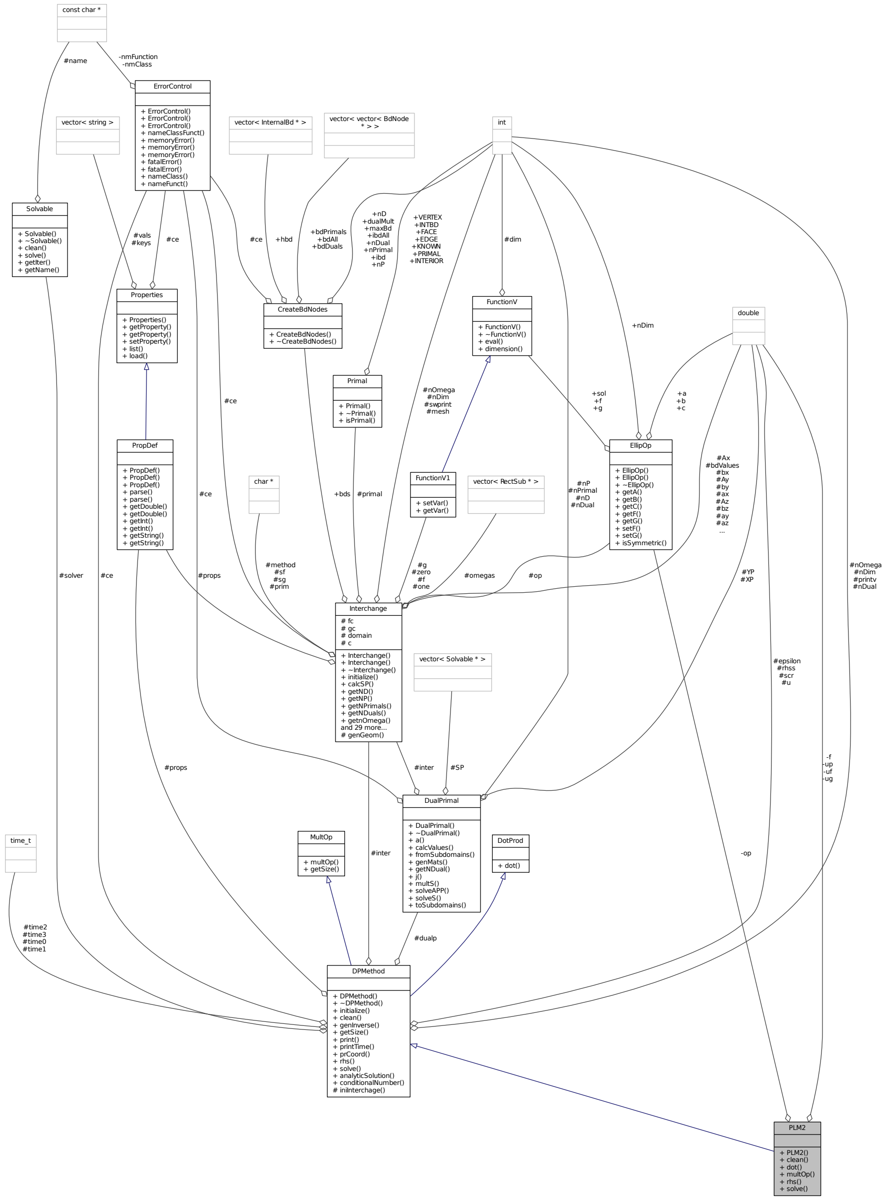
7.52 PLM2 Class Reference

`#include <PLM2.hpp>`

Inheritance diagram for PLM2:



Collaboration diagram for PLM2:



Public Member Functions

- PLM2 (PropDef &props, EllipOp &op)
- virtual void clean (void)
- ldouble dot (ldouble ∗u, ldouble ∗v)
- void multOp (ldouble ∗u, ldouble ∗v)

  y = A∗x
- void rhs (void)
- void solve (void)

Private Attributes

- ldouble ∗ up
- ldouble ∗ uf
- ldouble ∗ ug
- ldouble ∗ f
- EllipOp ∗ op

Additional Inherited Members

7.52.1 Constructor & Destructor Documentation

7.52.1.1 PLM2::PLM2 (PropDef & *props*, EllipOp & *op*) [inline]

7.52.2 Member Function Documentation

7.52.2.1 virtual void PLM2::clean (void) [inline], [virtual]

Implements DPMethod.

Reimplemented in PLM2MPI.

7.52.2.2 ldouble PLM2::dot (ldouble ∗ *u*, ldouble ∗ *v*) [virtual]

Implements DotProd.

7.52.2.3 void PLM2::multOp (ldouble ∗ *x*, ldouble ∗ *y*) [virtual]

y = A∗x

Implements MultOp.

7.52.2.4 void PLM2::rhs (void) [virtual]

Implements DPMethod.

7.52.2.5 void PLM2::solve (void) [virtual]

Implements DPMethod.

7.52.3 Member Data Documentation

7.52.3.1 ldouble∗ PLM2::f [private]

7.52.3.2 EllipOp∗ PLM2::op [private]

7.52.3.3 ldouble∗ PLM2::uf [private]

7.52.3.4 Idouble∗ PLM2::ug [private]

7.52.3.5 Idouble∗ PLM2::up [private]

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

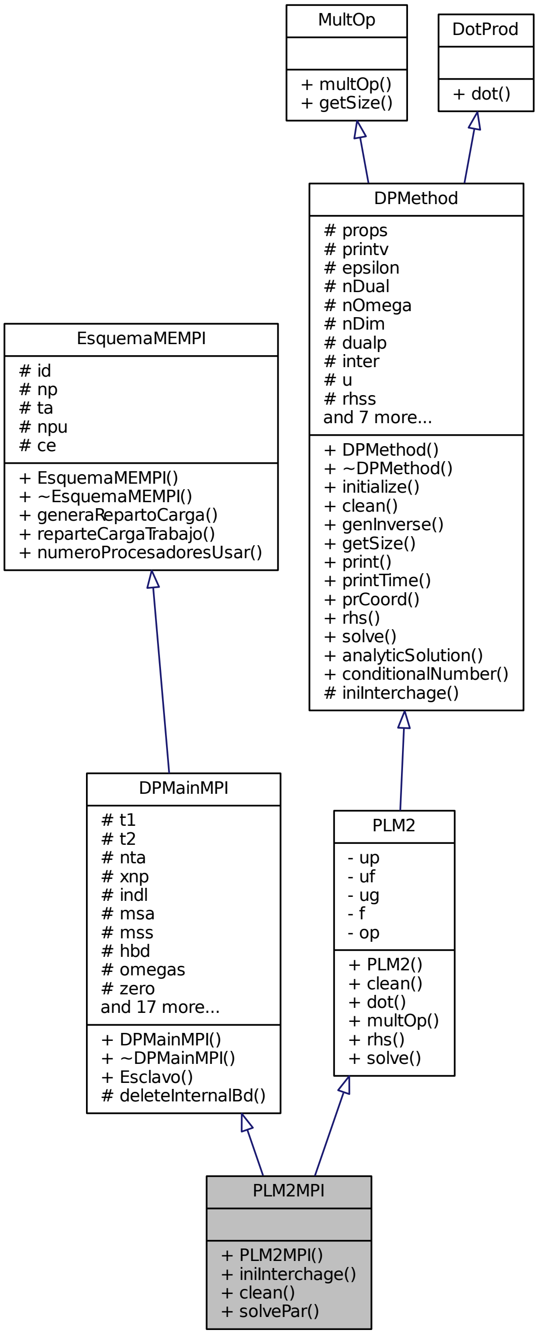
- PLM2.hpp
- PLM2.cpp

7.53 PLM2MPI Class Reference

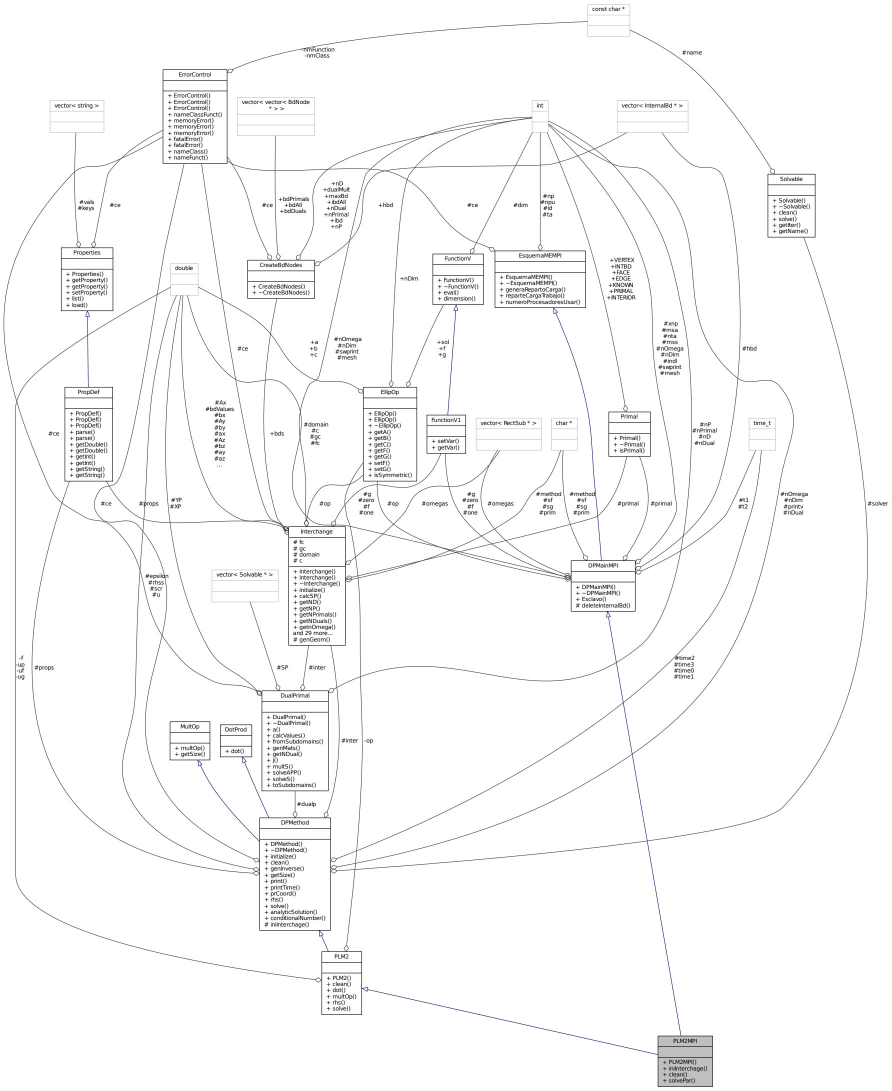
Clase para definir el metodo MF-1 de DVS-DDM.

```
#include <PLM2MPI.hpp>
```

Inheritance diagram for PLM2MPI:



Collaboration diagram for PLM2MPI:



Public Member Functions

- PLM2MPI (int id, int np, PropDef &props, EllipOp &op)

  Constructor de la clase.
- void iniInterchage (void)

  Inicializa InterchangeMPI en lugar de Interchange.
- void clean (void)
- void solvePar (void)

  Sobrecarga del la aplicacion.

Additional Inherited Members

7.53.1 Detailed Description

Clase para definir el metodo MF-1 de DVS-DDM.

Clase para definir el metodo MF-1 de DVS-DDM en paralelo

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.53.2 Constructor & Destructor Documentation

7.53.2.1 PLM2MPI::PLM2MPI (int *id*, int *np*, PropDef & *props*, EllipOp & *op*) `[inline]`

Constructor de la clase.

7.53.3 Member Function Documentation

7.53.3.1 void PLM2MPI::clean (void) `[inline], [virtual]`

Reimplemented from PLM2.

7.53.3.2 void PLM2MPI::iniInterchage (void) `[inline], [virtual]`

Inicializa InterchangeMPI en lugar de Interchange.

Reimplemented from DPMethod.

7.53.3.3 void PLM2MPI::solvePar (void) `[inline]`

Sobrecarga del la aplicacion.

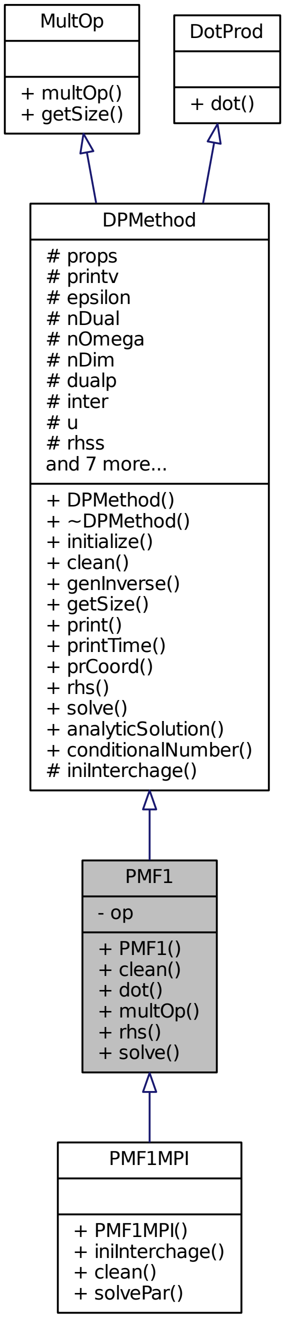
The documentation for this class was generated from the following file:

- PLM2MPI.hpp

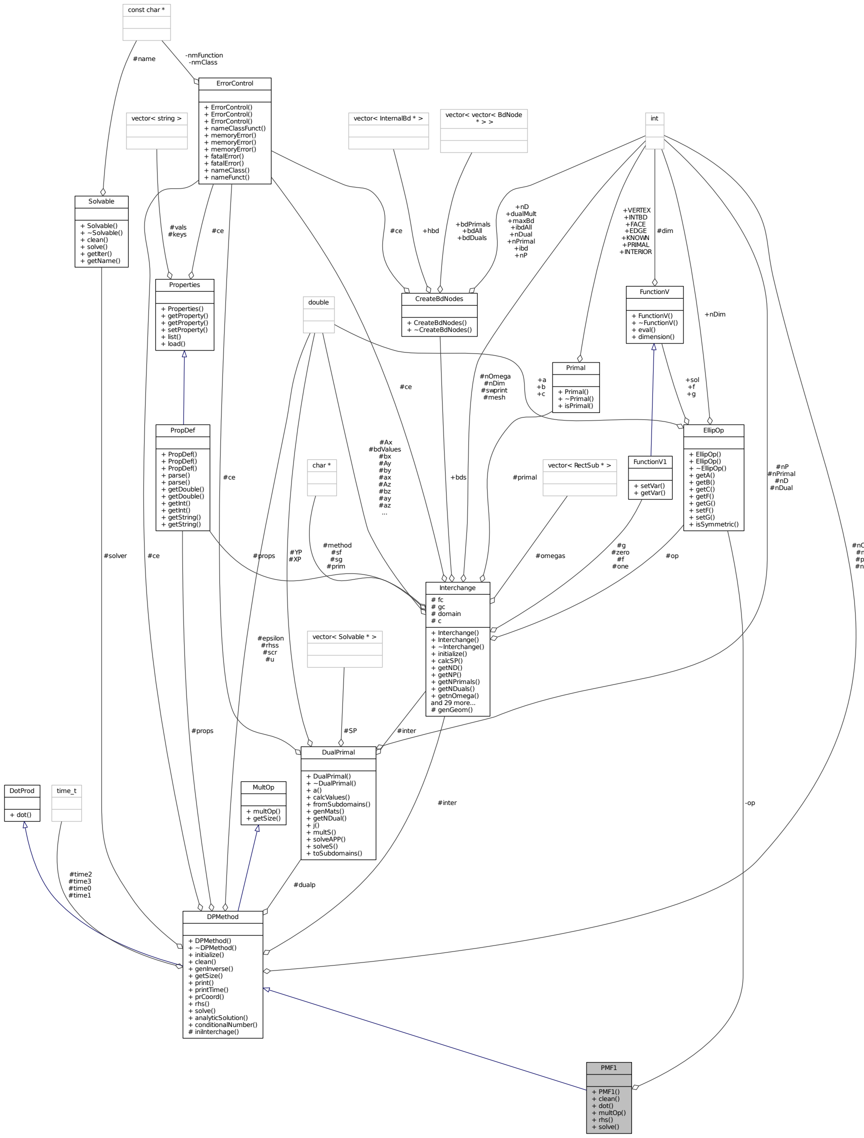
7.54 PMF1 Class Reference

```
#include <PMF1.hpp>
```

Inheritance diagram for PMF1:



Collaboration diagram for PMF1:



Public Member Functions

- PMF1 (PropDef &props, EllipOp &op)
- virtual void clean (void)
- ldouble dot (ldouble ∗u, ldouble ∗v)
- void multOp (ldouble ∗u, ldouble ∗v)

  $y = A*x$
- void rhs (void)
- void solve (void)

Private Attributes

- EllipOp ∗ op

Additional Inherited Members

7.54.1 Constructor & Destructor Documentation

7.54.1.1 PMF1::PMF1 (PropDef & *props*, EllipOp & *op*) [inline]

7.54.2 Member Function Documentation

7.54.2.1 virtual void PMF1::clean (void) [inline], [virtual]

Implements DPMethod.

Reimplemented in PMF1MPI.

7.54.2.2 ldouble PMF1::dot (ldouble ∗ *u*, ldouble ∗ *v*) [virtual]

Implements DotProd.

7.54.2.3 void PMF1::multOp (ldouble ∗ *x*, ldouble ∗ *y*) [virtual]

y = A∗x

Implements MultOp.

7.54.2.4 void PMF1::rhs (void) [virtual]

Implements DPMethod.

7.54.2.5 void PMF1::solve (void) [virtual]

Implements DPMethod.

7.54.3 Member Data Documentation

7.54.3.1 EllipOp∗ PMF1::op [private]

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

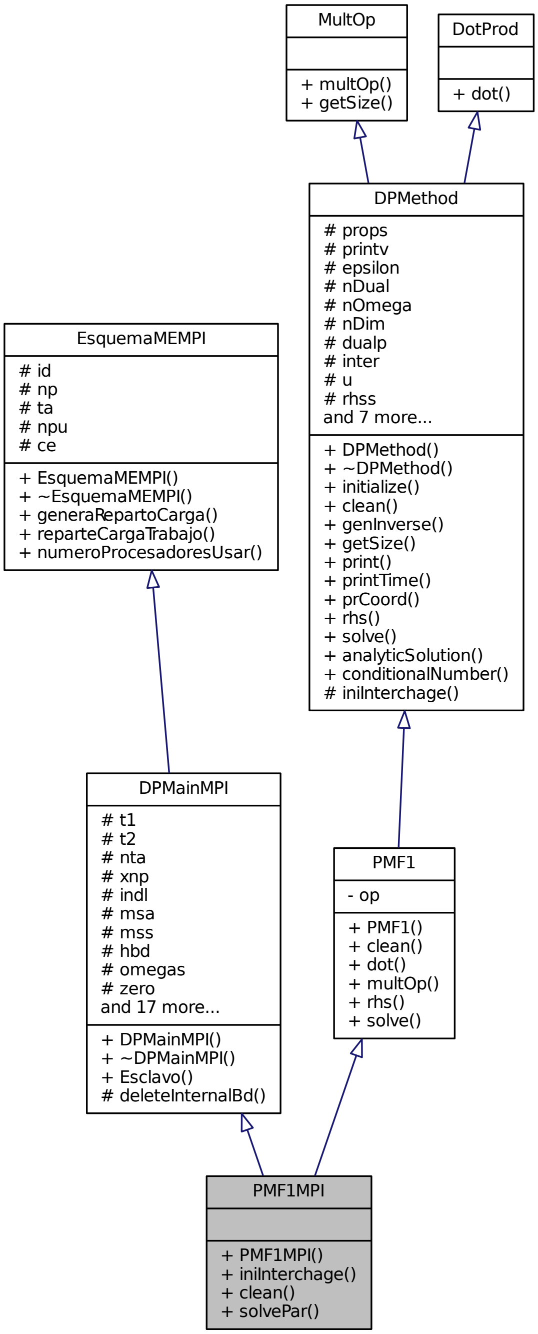
- PMF1.hpp
- PMF1.cpp

7.55 PMF1MPI Class Reference

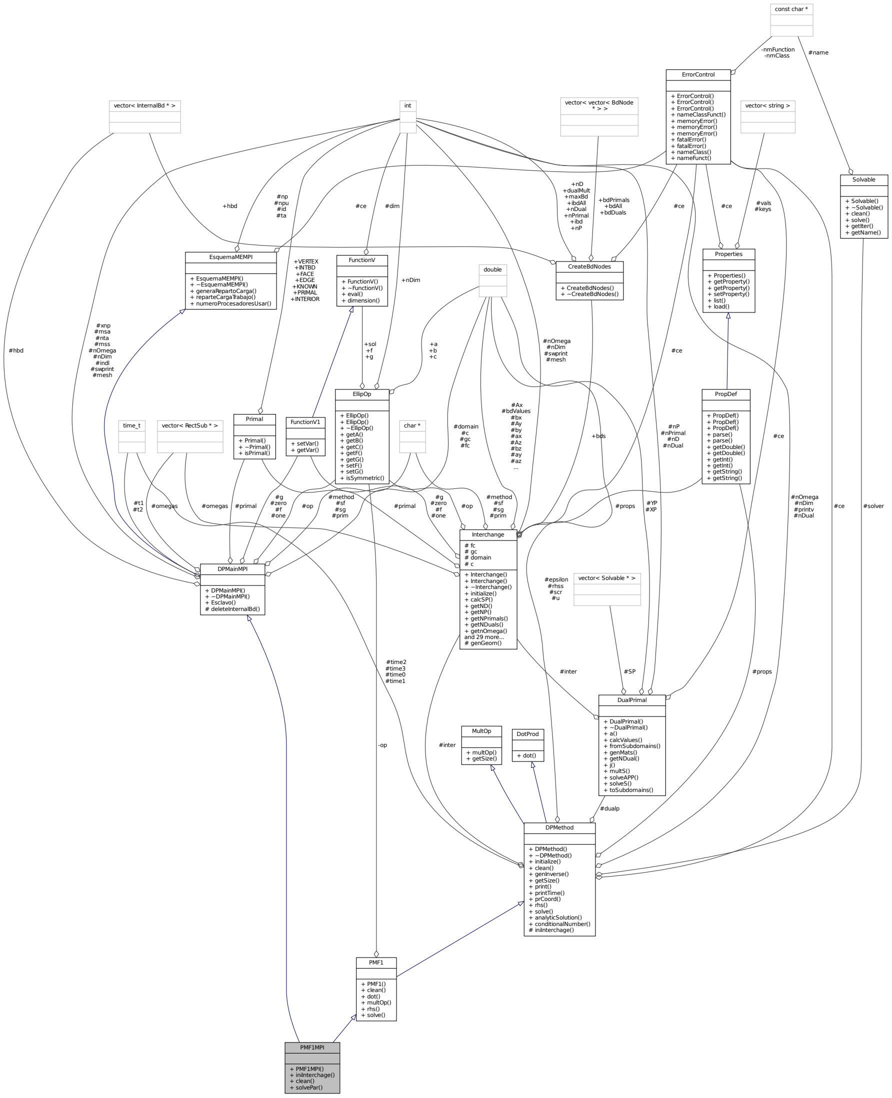
Clase para definir el metodo PMF-1 de DVS-DDM.

```
#include <PMF1MPI.hpp>
```

Inheritance diagram for PMF1MPI:



Collaboration diagram for PMF1MPI:



Public Member Functions

- PMF1MPI (int id, int np, PropDef &props, EllipOp &op)

    Constructor de la clase.

- void iniInterchage (void)

    Inicializa InterchangeMPI en lugar de Interchange.

- void clean (void)
- void solvePar (void)

    Sobrecarga del la aplicacion.

Additional Inherited Members

7.55.1 Detailed Description

Clase para definir el metodo PMF-1 de DVS-DDM.

Clase para definir el metodo PMF-1 de DVS-DDM en paralelo

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.55.2 Constructor & Destructor Documentation

7.55.2.1 PMF1MPI::PMF1MPI (int *id,* int *np,* PropDef & *props,* EllipOp & *op*) `[inline]`

Constructor de la clase.

7.55.3 Member Function Documentation

7.55.3.1 void PMF1MPI::clean (void) `[inline], [virtual]`

Reimplemented from PMF1.

7.55.3.2 void PMF1MPI::iniInterchage (void) `[inline], [virtual]`

Inicializa InterchangeMPI en lugar de Interchange.

Reimplemented from DPMethod.

7.55.3.3 void PMF1MPI::solvePar (void) `[inline]`

Sobrecarga del la aplicacion.

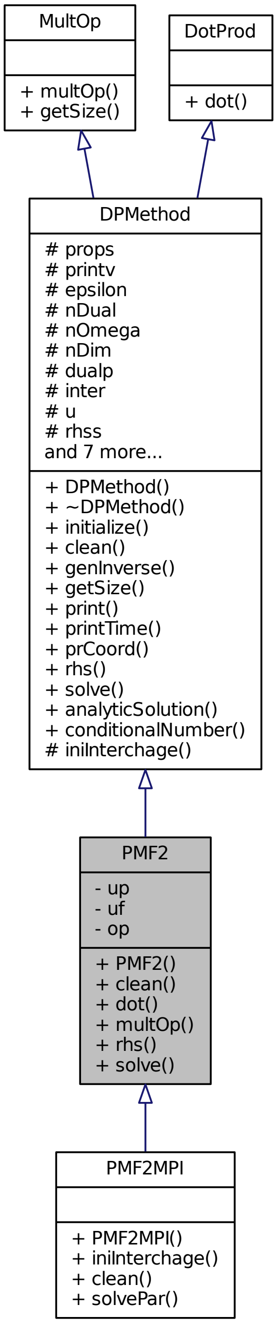
The documentation for this class was generated from the following file:

- PMF1MPI.hpp

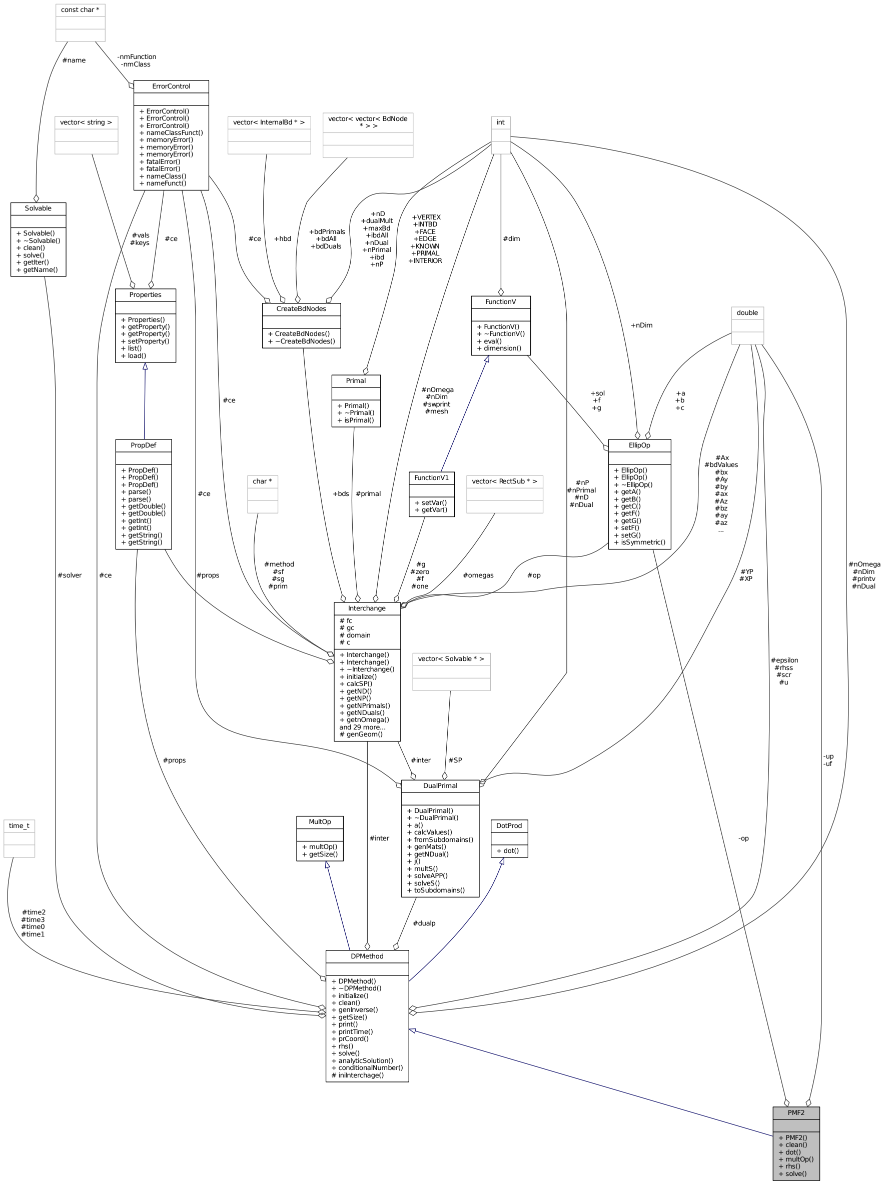
7.56 PMF2 Class Reference

```
#include <PMF2.hpp>
```

Inheritance diagram for PMF2:



Collaboration diagram for PMF2:



Public Member Functions

- PMF2 (PropDef &props, EllipOp &op)
- virtual void clean (void)
- ldouble dot (ldouble ∗u, ldouble ∗v)
- void multOp (ldouble ∗u, ldouble ∗v)

  $y = A*x$
- void rhs (void)
- void solve (void)

Private Attributes

- ldouble ∗ up
- ldouble ∗ uf
- EllipOp ∗ op

Additional Inherited Members

7.56.1 Constructor & Destructor Documentation

7.56.1.1 PMF2::PMF2 (PropDef & *props*, EllipOp & *op*) [inline]

7.56.2 Member Function Documentation

7.56.2.1 virtual void PMF2::clean (void) [inline], [virtual]

Implements DPMethod.

Reimplemented in PMF2MPI.

7.56.2.2 ldouble PMF2::dot (ldouble ∗ *u*, ldouble ∗ *v*) [virtual]

Implements DotProd.

7.56.2.3 void PMF2::multOp (ldouble ∗ *x*, ldouble ∗ *y*) [virtual]

y = A∗x

Implements MultOp.

7.56.2.4 void PMF2::rhs (void) [virtual]

Implements DPMethod.

7.56.2.5 void PMF2::solve (void) [virtual]

Implements DPMethod.

7.56.3 Member Data Documentation

7.56.3.1 EllipOp∗ PMF2::op [private]

7.56.3.2 ldouble∗ PMF2::uf [private]

7.56.3.3 ldouble∗ PMF2::up [private]

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

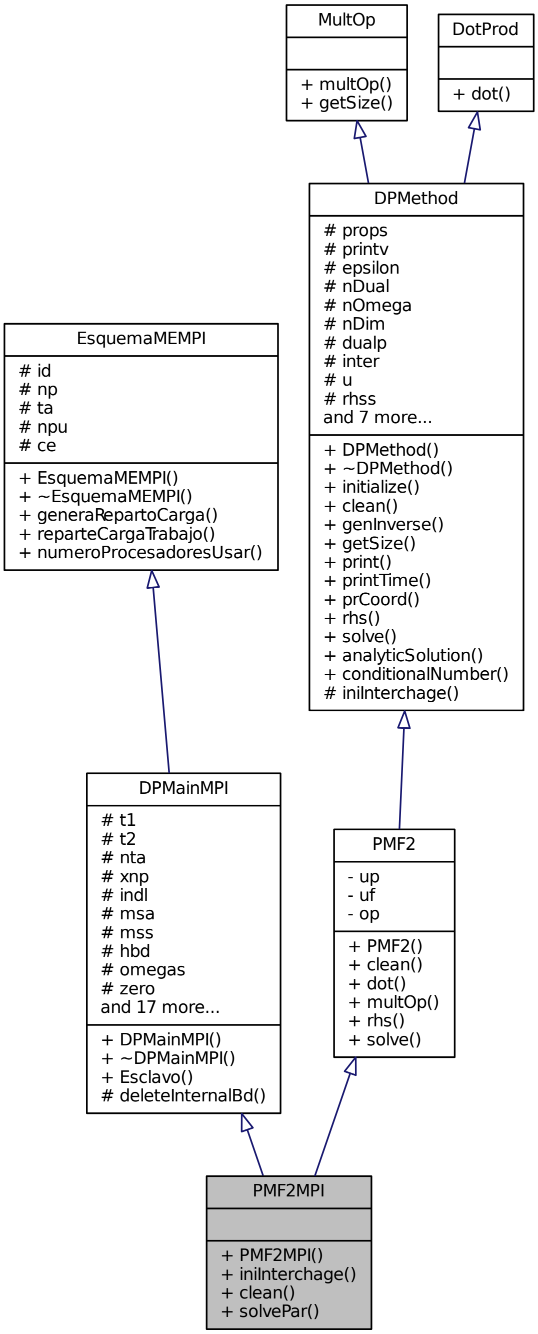
- PMF2.hpp
- PMF2.cpp

7.57 PMF2MPI Class Reference

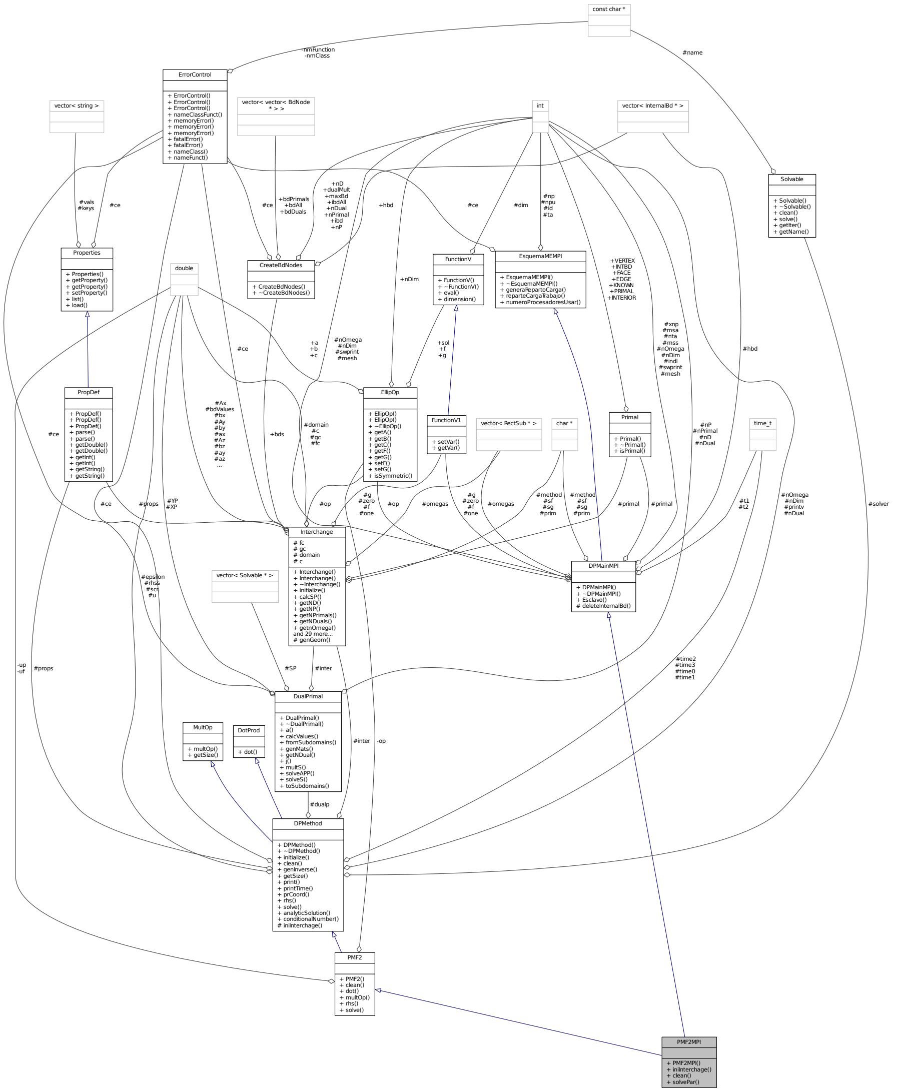
Clase para definir el metodo PMF-2 de DVS-DDM.

`#include <PMF2MPI.hpp>`

Inheritance diagram for PMF2MPI:



Collaboration diagram for PMF2MPI:



Public Member Functions

- PMF2MPI (int id, int np, PropDef &props, EllipOp &op)

  Constructor de la clase.

- void iniInterchage (void)

  Inicializa InterchangeMPI en lugar de Interchange.

- void clean (void)
- void solvePar (void)

  Sobrecarga del la aplicacion.

Additional Inherited Members

7.57.1 Detailed Description

Clase para definir el metodo PMF-2 de DVS-DDM.

Clase para definir el metodo PMF-2 de DVS-DDM en paralelo

Author

Antonio Carrillo Ledesma

Date

primavera 2010

Version

1.0.0

Bug No hay errores conocidos

7.57.2 Constructor & Destructor Documentation

7.57.2.1 PMF2MPI::PMF2MPI (int *id,* int *np,* PropDef & *props,* EllipOp & *op*) `[inline]`

Constructor de la clase.

7.57.3 Member Function Documentation

7.57.3.1 void PMF2MPI::clean (void) `[inline], [virtual]`

Reimplemented from PMF2.

7.57.3.2 void PMF2MPI::iniInterchage (void) `[inline], [virtual]`

Inicializa InterchangeMPI en lugar de Interchange.

Reimplemented from DPMethod.

7.57.3.3 void PMF2MPI::solvePar (void) `[inline]`

Sobrecarga del la aplicacion.

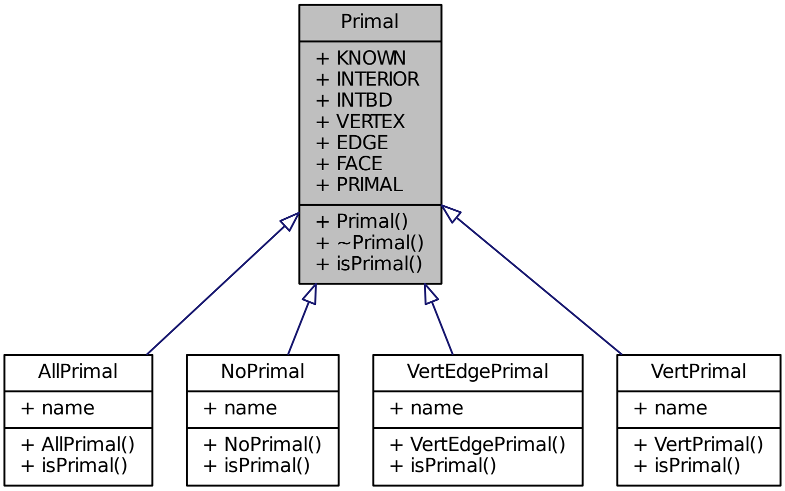
The documentation for this class was generated from the following file:

- PMF2MPI.hpp

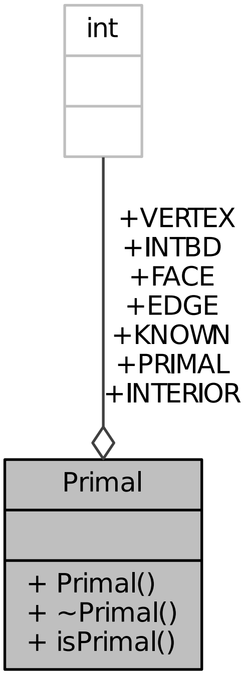
7.58 Primal Class Reference

`#include <Primal.hpp>`

Inheritance diagram for Primal:



Collaboration diagram for Primal:



Public Member Functions

- Primal ()
- virtual ∼Primal ()
- virtual bool isPrimal (int type, int ∗coordN, int ∗coordM)=0

Static Public Attributes

- static const int KNOWN = 1
- static const int INTERIOR = 2
- static const int INTBD = 4
- static const int VERTEX = 8
- static const int EDGE = 16
- static const int FACE = 32
- static const int PRIMAL = 64

7.58.1 Constructor & Destructor Documentation

7.58.1.1 Primal::Primal ( ) [inline]

7.58.1.2 virtual Primal::∼Primal ( ) [inline], [virtual]

7.58.2 Member Function Documentation

7.58.2.1 virtual bool Primal::isPrimal (int *type*, int ∗ *coordN*, int ∗ *coordM*) [pure virtual]

Implemented in AllPrimal, NoPrimal, VertEdgePrimal, and VertPrimal.

7.58.3 Member Data Documentation

7.58.3.1 const int Primal::EDGE = 16 [static]

7.58.3.2 const int Primal::FACE = 32 [static]

7.58.3.3 const int Primal::INTBD = 4 [static]

7.58.3.4 const int Primal::INTERIOR = 2 [static]

7.58.3.5 const int Primal::KNOWN = 1 [static]

7.58.3.6 const int Primal::PRIMAL = 64 [static]

7.58.3.7 const int Primal::VERTEX = 8 [static]

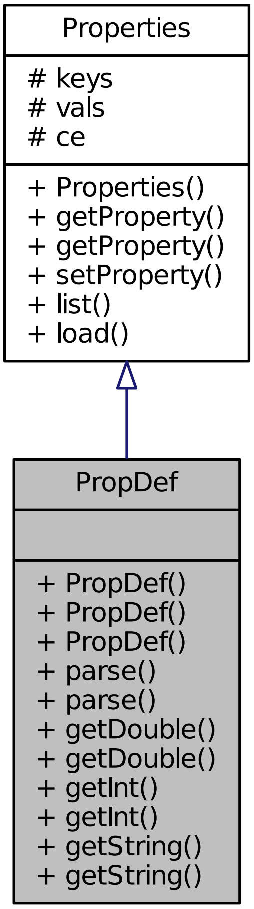
The documentation for this class was generated from the following file:

- Primal.hpp

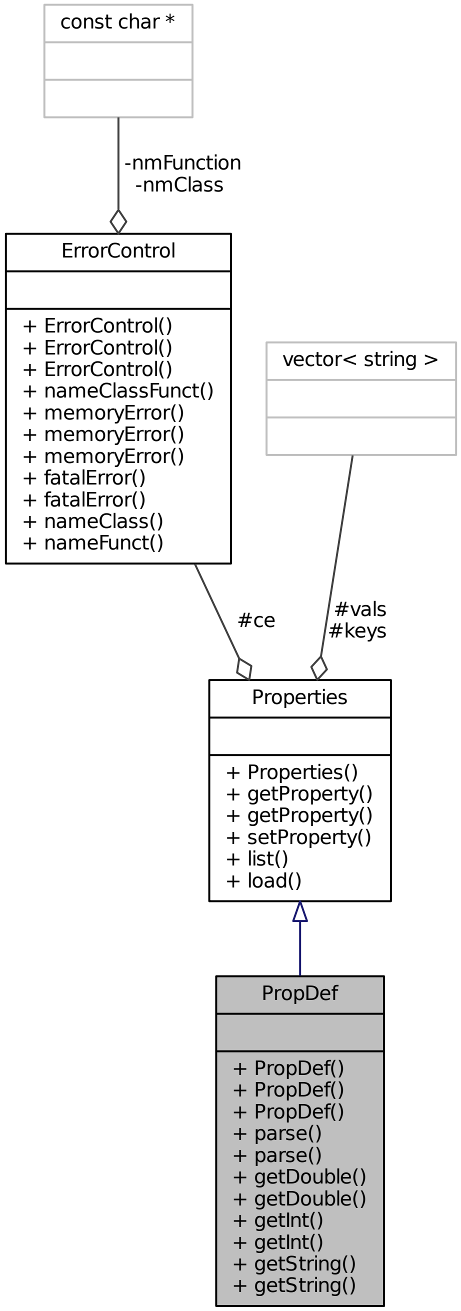
7.59 PropDef Class Reference

```
#include <PropDef.hpp>
```

Inheritance diagram for PropDef:



Collaboration diagram for PropDef:



Public Member Functions

- PropDef (void)
- PropDef (Properties prop)

- PropDef (int nargs, char ∗args[ ])
- int parse (string &file)
- int parse (int nargs, char ∗args[ ])
- ldouble getDouble (const char ∗key, ldouble value)
- ldouble getDouble (const char ∗key)
- int getInt (const char ∗key, int value)
- int getInt (const char ∗key)
- char ∗ getString (const char ∗key, const char ∗value)
- const char ∗ getString (const char ∗key)

Additional Inherited Members

7.59.1 Constructor & Destructor Documentation

7.59.1.1 PropDef::PropDef (void) [inline]

7.59.1.2 PropDef::PropDef (Properties *prop*) [inline]

7.59.1.3 PropDef::PropDef (int *nargs,* char ∗ *args[ ]*) [inline]

7.59.2 Member Function Documentation

7.59.2.1 ldouble PropDef::getDouble (const char ∗ *key,* ldouble *value*)

7.59.2.2 ldouble PropDef::getDouble (const char ∗ *key*)

7.59.2.3 int PropDef::getInt (const char ∗ *key,* int *value*)

7.59.2.4 int PropDef::getInt (const char ∗ *key*)

7.59.2.5 char ∗ PropDef::getString (const char ∗ *key,* const char ∗ *value*)

7.59.2.6 const char ∗ PropDef::getString (const char ∗ *key*)

7.59.2.7 int PropDef::parse (string & *file*)

7.59.2.8 int PropDef::parse (int *nargs,* char ∗ *args[ ]*)

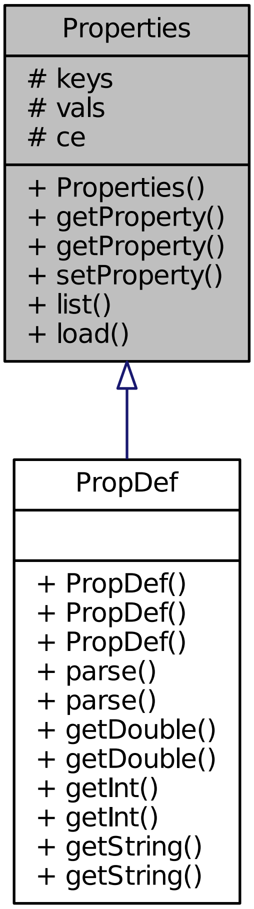
The documentation for this class was generated from the following files:

- PropDef.hpp
- PropDef.cpp

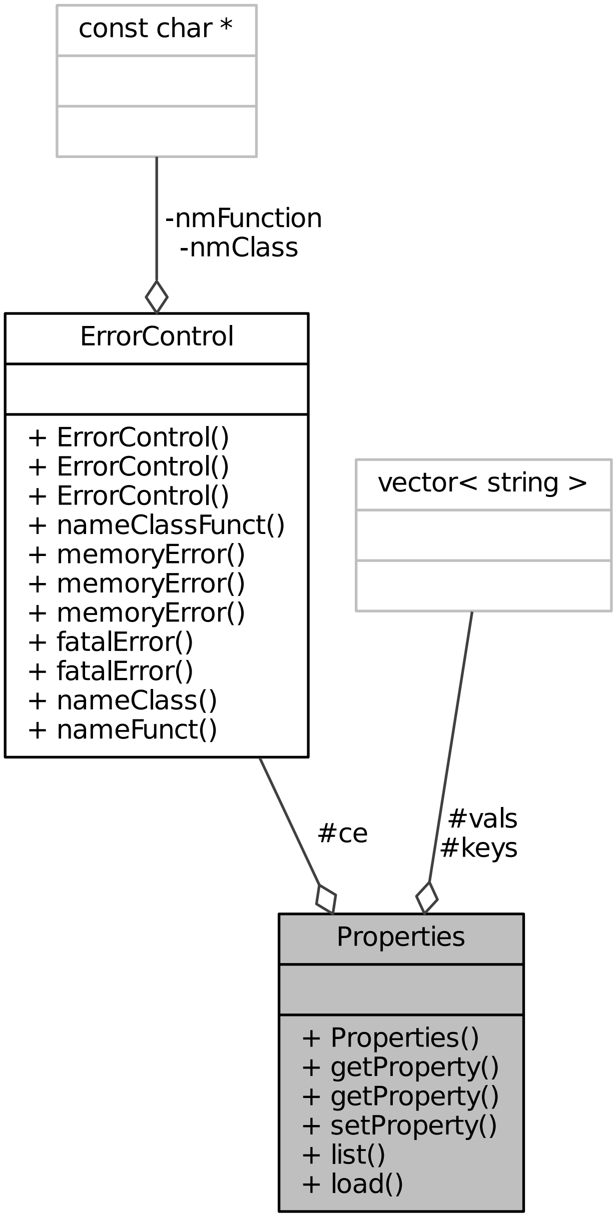
7.60 Properties Class Reference

`#include <Properties.hpp>`

Inheritance diagram for Properties:



Collaboration diagram for Properties:



Public Member Functions

- Properties (void)
- char ∗ getProperty (const char ∗s, const char ∗val)
- const char ∗ getProperty (const char ∗s)

- const char ∗ setProperty (const char ∗k, const char ∗v)
- void list (void)
- void load (istream &stream)

Protected Attributes

- vector< string > keys
- vector< string > vals
- ErrorControl ce

  Control de errores.

7.60.1 Constructor & Destructor Documentation

7.60.1.1 Properties::Properties (void) [inline]

7.60.2 Member Function Documentation

7.60.2.1 char ∗ Properties::getProperty (const char ∗ *s*, const char ∗ *val*)

7.60.2.2 const char∗ Properties::getProperty (const char ∗ *s*) [inline]

7.60.2.3 void Properties::list (void)

7.60.2.4 void Properties::load (istream & *stream*)

7.60.2.5 const char ∗ Properties::setProperty (const char ∗ *k*, const char ∗ *v*)

7.60.3 Member Data Documentation

7.60.3.1 ErrorControl Properties::ce [protected]

Control de errores.

7.60.3.2 vector<string> Properties::keys [protected]

7.60.3.3 vector<string> Properties::vals [protected]

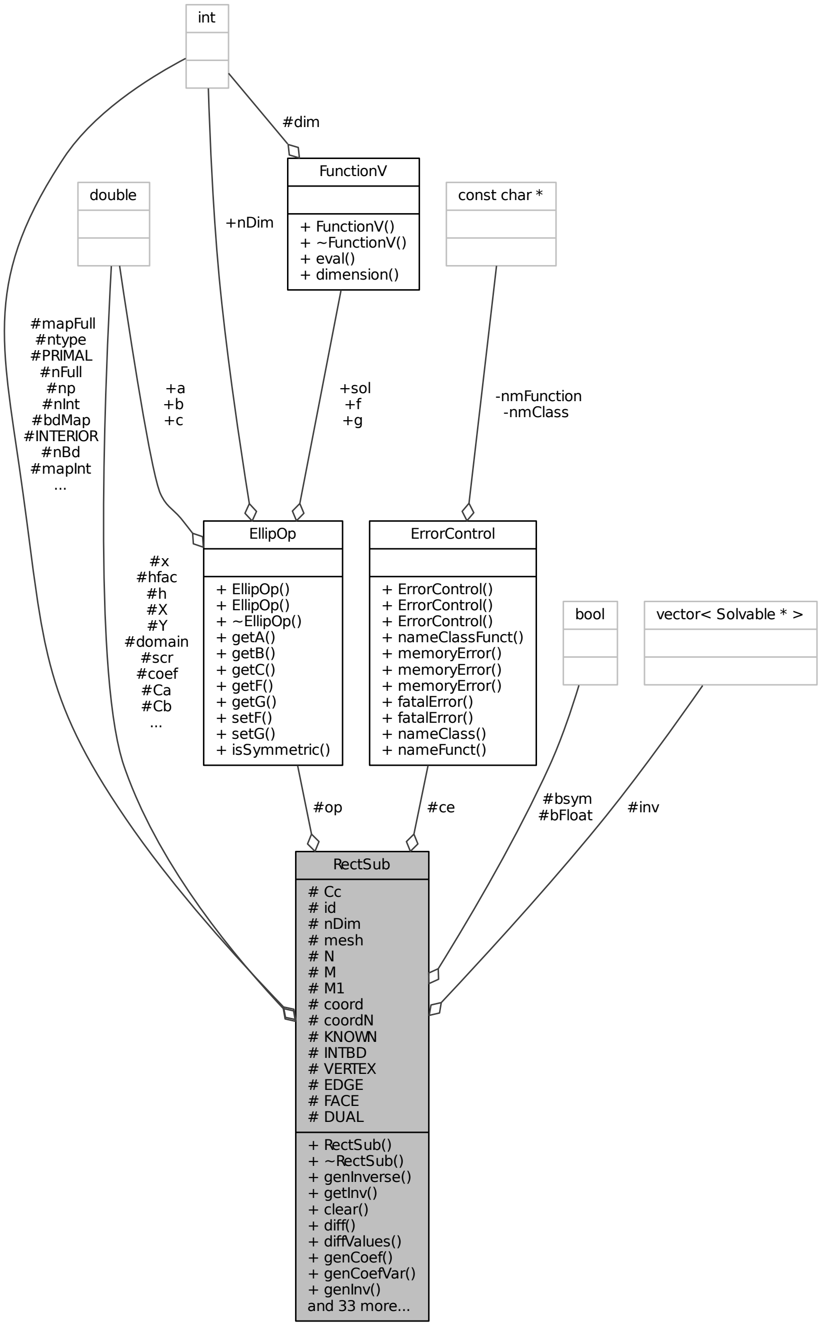
The documentation for this class was generated from the following files:

- Properties.hpp
- Properties.cpp

7.61 RectSub Class Reference

```
#include <RectSub.hpp>
```

Collaboration diagram for RectSub:



Public Member Functions

- RectSub (int id, int nDim, int ∗mesh, ldouble ∗∗dom, EllipOp &op, Primal &primal)
- ∼RectSub (void)

- Solvable ∗ genInverse (int ∗map, ldouble fac)
- vector< Solvable ∗ > getInv (void)
- void clear (int s)
- void diff (int sc3, int sc1, int sc2)
- void diffValues (int sc, ldouble ∗u)
- void genCoef (EllipOp &op)
- void genCoefVar (int ren)
- void genInv (int type)
- void genNcoord (int n, int ∗coord, int ∗N)
- void genNtype (Primal &primal)
- int getBdSize ()
- void getCoord (int m, ldouble ∗x)
- void getCoordNode (int n, ldouble ∗x)
- vector< InternalBd ∗ > getInternalBd (void)
- int ∗ getNtype (void)
- void setNtype (int ∗arr)
- void getPrimals (int sc, ldouble ∗u)
- ldouble getValue (int sc, int n)
- void getValues (int sc, ldouble ∗u)
- void inverse (int sp, int sc1, int sc2)
- bool isKnown (int ∗coord)
- bool isInterior (int ∗coord)
- bool isIntBd (int ∗coord)
- int nodeType (int ∗coord)
- bool isDual (int i)
- bool isFloat (void)
- bool isInterior (int i)
- bool isKnown (int i)
- bool isPrimal (int i)
- bool isVertex (int i)
- void knownValues (int s1)
- void multOp (int s1, int s2)
- void printMat (const char ∗s, ldouble ∗∗A, int tm)
- void printMult (void)
- void rhs (int sc)
- void setPrimals (int sc, ldouble ∗u)
- void setValue (int sc, int n, ldouble val)
- void setValues (int sc, ldouble ∗u)
- void print (const char ∗s, int sc)
- void print (int sc)
- int getNP (void)

Protected Attributes

- ldouble Ca [3]
- ldouble Cb [3]
- ldouble Cc
- vector< Solvable ∗ > inv
- ldouble ∗ X
- ldouble ∗ Y

- int id
- int nDim
- ldouble ∗∗ domain
- EllipOp ∗ op
- int ∗ mesh
- int ∗ N
- int ∗ M
- int ∗ M1
- int ∗ coord
- int ∗ coordN
- ldouble ∗ h
- ldouble hfac
- ldouble ∗∗ scr
- int np
- int ∗ ntype
- ldouble ∗ coef
- int ∗ bdMap
- int ∗ mapInt
- int ∗ mapFull
- int nInt
- int nFull
- bool bFloat
- bool bsym
- ldouble ∗ x
- int nBd
- ErrorControl ce

    Control de errores.

Static Protected Attributes

- static const int KNOWN = 1
- static const int INTERIOR = 2
- static const int INTBD = 4
- static const int VERTEX = 8
- static const int EDGE = 16
- static const int FACE = 32
- static const int PRIMAL = 64
- static const int DUAL = 128

7.61.1 Constructor & Destructor Documentation

7.61.1.1 RectSub::RectSub (int *id*, int *nDim*, int ∗ *mesh*, ldouble ∗∗ *dom*, EllipOp & *op*, Primal & *primal*)

7.61.1.2 RectSub::∼RectSub (void) [inline]

7.61.2 Member Function Documentation

7.61.2.1 void RectSub::clear (int *s*)

7.61.2.2 void RectSub::diff (int sc3, int sc1, int sc2)

7.61.2.3 void RectSub::diffValues (int sc, Idouble * u) [inline]

7.61.2.4 void RectSub::genCoef (EllipOp & op)

7.61.2.5 void RectSub::genCoefVar (int ren)

7.61.2.6 void RectSub::genInv (int type)

7.61.2.7 Solvable * RectSub::genInverse (int * map, Idouble fac)

7.61.2.8 void RectSub::genNcoord (int n, int * coord, int * N)

7.61.2.9 void RectSub::genNtype (Primal & primal)

7.61.2.10 int RectSub::getBdSize ( ) [inline]

7.61.2.11 void RectSub::getCoord (int m, Idouble * x)

7.61.2.12 void RectSub::getCoordNode (int n, Idouble * x) [inline]

7.61.2.13 vector< InternalBd * > RectSub::getInternalBd (void)

7.61.2.14 vector<Solvable*> RectSub::getInv (void) [inline]

7.61.2.15 int RectSub::getNP (void) [inline]

7.61.2.16 int* RectSub::getNtype (void) [inline]

7.61.2.17 void RectSub::getPrimals (int sc, Idouble * u)

7.61.2.18 Idouble RectSub::getValue (int sc, int n) [inline]

7.61.2.19 void RectSub::getValues (int sc, Idouble * u)

7.61.2.20 void RectSub::inverse (int sp, int sc1, int sc2)

7.61.2.21 bool RectSub::isDual (int i) [inline]

7.61.2.22 bool RectSub::isFloat (void) [inline]

7.61.2.23 bool RectSub::isIntBd (int * coord)

7.61.2.24 bool RectSub::isInterior (int * coord)

7.61.2.25 bool RectSub::isInterior (int i) [inline]

7.61.2.26 bool RectSub::isKnown (int * coord)

7.61.2.27 bool RectSub::isKnown (int i) [inline]

7.61.2.28 bool RectSub::isPrimal (int *i*) [inline]

7.61.2.29 bool RectSub::isVertex (int *i*) [inline]

7.61.2.30 void RectSub::knownValues (int *s1*)

7.61.2.31 void RectSub::multOp (int *s1*, int *s2*)

7.61.2.32 int RectSub::nodeType (int * *coord*)

7.61.2.33 void RectSub::print (const char * *s*, int *sc*)

7.61.2.34 void RectSub::print (int *sc*)

7.61.2.35 void RectSub::printMat (const char * *s*, Idouble ** *A*, int *tm*)

7.61.2.36 void RectSub::printMult (void)

7.61.2.37 void RectSub::rhs (int *sc*)

7.61.2.38 void RectSub::setNtype (int * *arr*) [inline]

7.61.2.39 void RectSub::setPrimals (int *sc*, Idouble * *u*)

7.61.2.40 void RectSub::setValue (int *sc*, int *n*, Idouble *val*) [inline]

7.61.2.41 void RectSub::setValues (int *sc*, Idouble * *u*)

7.61.3 Member Data Documentation

7.61.3.1 int* RectSub::bdMap [protected]

7.61.3.2 bool RectSub::bFloat [protected]

7.61.3.3 bool RectSub::bsym [protected]

7.61.3.4 Idouble RectSub::Ca[3] [protected]

7.61.3.5 Idouble RectSub::Cb[3] [protected]

7.61.3.6 Idouble RectSub::Cc [protected]

7.61.3.7 ErrorControl RectSub::ce [protected]

Control de errores.

7.61.3.8 Idouble* RectSub::coef [protected]

7.61.3.9 int* RectSub::coord [protected]

7.61.3.10 int* RectSub::coordN [protected]

7.61.3.11 Idouble** RectSub::domain [protected]

7.61.3.12 const int RectSub::DUAL = 128 [static], [protected]

7.61.3.13 const int RectSub::EDGE = 16 [static], [protected]

7.61.3.14 const int RectSub::FACE = 32 [static], [protected]

7.61.3.15 Idouble* RectSub::h [protected]

7.61.3.16 Idouble RectSub::hfac [protected]

7.61.3.17 int RectSub::id [protected]

7.61.3.18 const int RectSub::INTBD = 4 [static], [protected]

7.61.3.19 const int RectSub::INTERIOR = 2 [static], [protected]

7.61.3.20 vector<Solvable*> RectSub::inv [protected]

7.61.3.21 const int RectSub::KNOWN = 1 [static], [protected]

7.61.3.22 int* RectSub::M [protected]

7.61.3.23 int* RectSub::M1 [protected]

7.61.3.24 int* RectSub::mapFull [protected]

7.61.3.25 int* RectSub::mapInt [protected]

7.61.3.26 int* RectSub::mesh [protected]

7.61.3.27 int* RectSub::N [protected]

7.61.3.28 int RectSub::nBd [protected]

7.61.3.29 int RectSub::nDim [protected]

7.61.3.30 int RectSub::nFull [protected]

7.61.3.31 int RectSub::nInt [protected]

7.61.3.32 int RectSub::np [protected]

7.61.3.33 int* RectSub::ntype [protected]

7.61.3.34 EllipOp* RectSub::op [protected]

7.61.3.35 const int RectSub::PRIMAL = 64 [static], [protected]

7.61.3.36 Idouble** RectSub::scr [protected]

7.61.3.37 const int RectSub::VERTEX = 8 [static], [protected]

7.61.3.38 Idouble∗ RectSub::X [protected]

7.61.3.39 Idouble∗ RectSub::x [protected]

7.61.3.40 Idouble∗ RectSub::Y [protected]

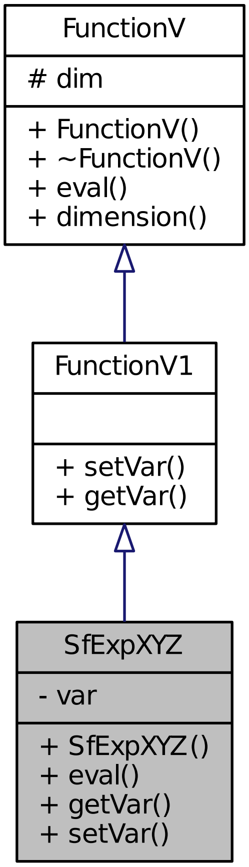
The documentation for this class was generated from the following files:

- RectSub.hpp
- RectSub.cpp

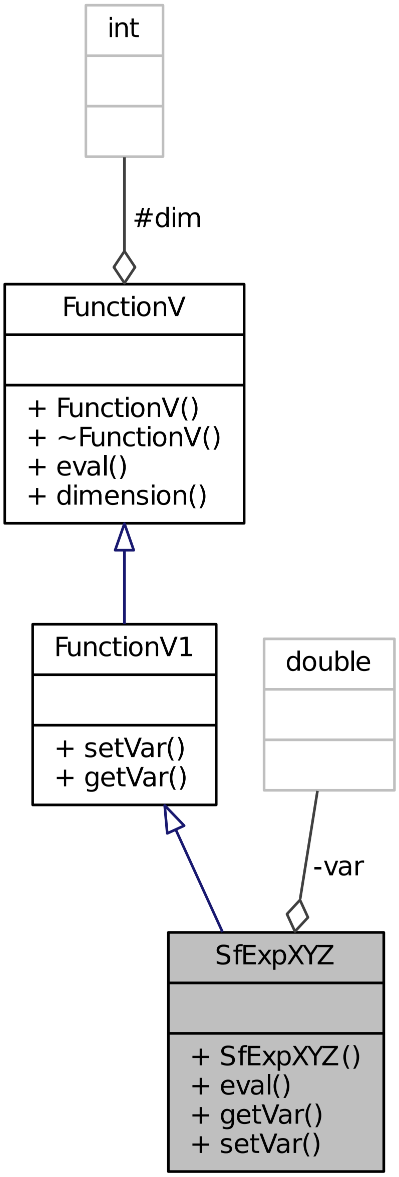
7.62 SfExpXYZ Class Reference

```
#include <SfExpXYZ.hpp>
```

Inheritance diagram for SfExpXYZ:



Collaboration diagram for SfExpXYZ:



Public Member Functions

- SfExpXYZ (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.62.1 Constructor & Destructor Documentation

7.62.1.1 SfExpXYZ::SfExpXYZ (**Idouble** ***b***) `[inline]`

7.62.2 Member Function Documentation

7.62.2.1 **Idouble** SfExpXYZ::eval (int ***d,*** **Idouble** ∗ ***x***) `[inline], [virtual]`

Implements FunctionV.

7.62.2.2 **Idouble** SfExpXYZ::getVar (void) `[inline], [virtual]`

Implements FunctionV1.

7.62.2.3 void SfExpXYZ::setVar (**Idouble** ***b***) `[inline], [virtual]`

Implements FunctionV1.

7.62.3 Member Data Documentation

7.62.3.1 **Idouble** SfExpXYZ::var `[private]`

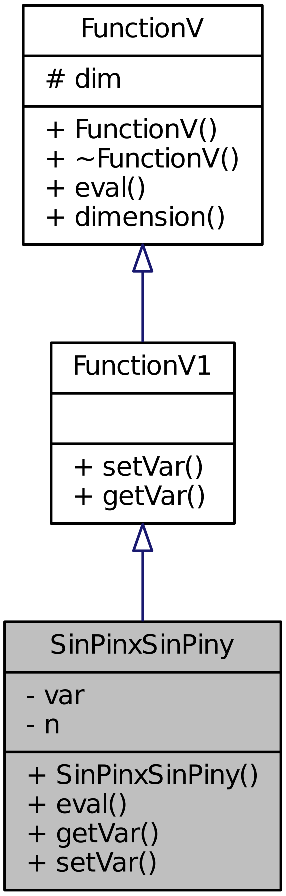
The documentation for this class was generated from the following file:

- SfExpXYZ.hpp

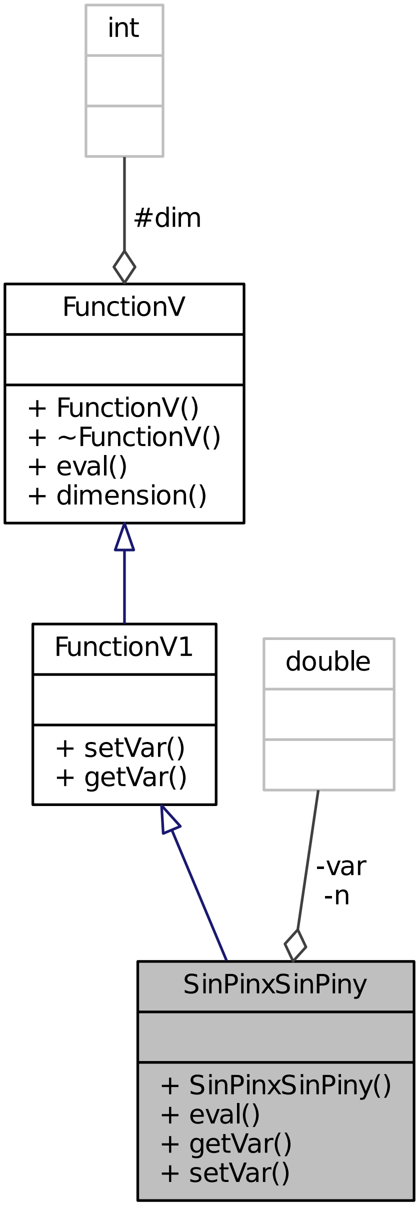
7.63 SinPinxSinPiny Class Reference

```
#include <SinPinxSinPiny.hpp>
```

Inheritance diagram for SinPinxSinPiny:



Collaboration diagram for SinPinxSinPiny:



Public Member Functions

- SinPinxSinPiny (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)

- void setVar (ldouble b)

Private Attributes

- ldouble var
- ldouble n

Additional Inherited Members

7.63.1 Constructor & Destructor Documentation

7.63.1.1 SinPinxSinPiny::SinPinxSinPiny (ldouble *b*) `[inline]`

7.63.2 Member Function Documentation

7.63.2.1 ldouble SinPinxSinPiny::eval (int *d*, ldouble ∗ *x*) `[inline]`, `[virtual]`

Implements FunctionV.

7.63.2.2 ldouble SinPinxSinPiny::getVar (void) `[inline]`, `[virtual]`

Implements FunctionV1.

7.63.2.3 void SinPinxSinPiny::setVar (ldouble *b*) `[inline]`, `[virtual]`

Implements FunctionV1.

7.63.3 Member Data Documentation

7.63.3.1 ldouble SinPinxSinPiny::n `[private]`

7.63.3.2 ldouble SinPinxSinPiny::var `[private]`

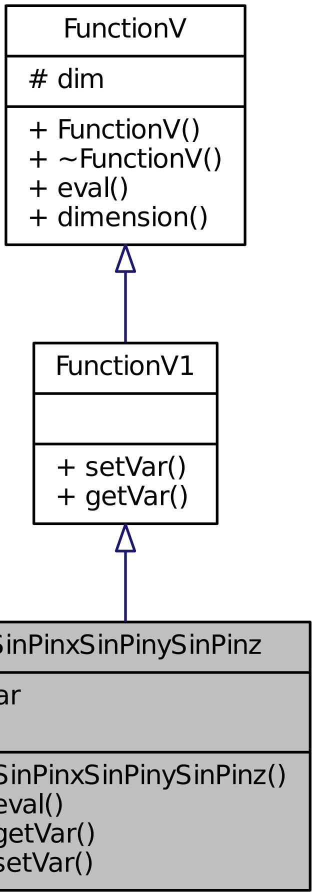
The documentation for this class was generated from the following file:

- SinPinxSinPiny.hpp

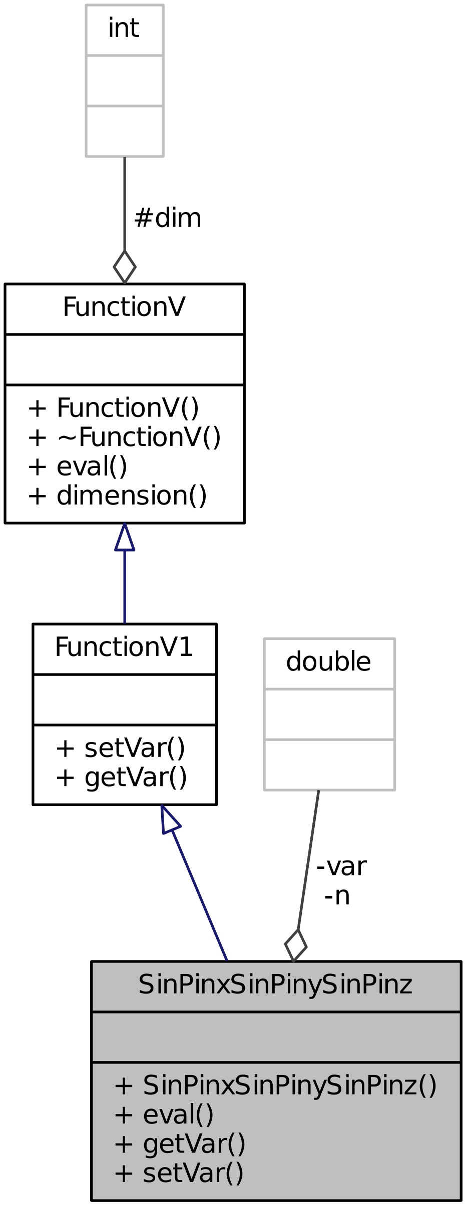
7.64 SinPinxSinPinySinPinz Class Reference

```
#include <SinPinxSinPinySinPinz.hpp>
```

Inheritance diagram for SinPinxSinPinySinPinz:



Collaboration diagram for SinPinxSinPinySinPinz:



Public Member Functions

- SinPinxSinPinySinPinz (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)

- void setVar (ldouble b)

Private Attributes

- ldouble var
- ldouble n

Additional Inherited Members

7.64.1 Constructor & Destructor Documentation

7.64.1.1 SinPinxSinPinySinPinz::SinPinxSinPinySinPinz (**ldouble *b***) `[inline]`

7.64.2 Member Function Documentation

7.64.2.1 ldouble SinPinxSinPinySinPinz::eval (int *d*, ldouble ∗ *x*) `[inline], [virtual]`

Implements FunctionV.

7.64.2.2 ldouble SinPinxSinPinySinPinz::getVar (void) `[inline], [virtual]`

Implements FunctionV1.

7.64.2.3 void SinPinxSinPinySinPinz::setVar (**ldouble *b***) `[inline], [virtual]`

Implements FunctionV1.

7.64.3 Member Data Documentation

7.64.3.1 ldouble SinPinxSinPinySinPinz::n `[private]`

7.64.3.2 ldouble SinPinxSinPinySinPinz::var `[private]`

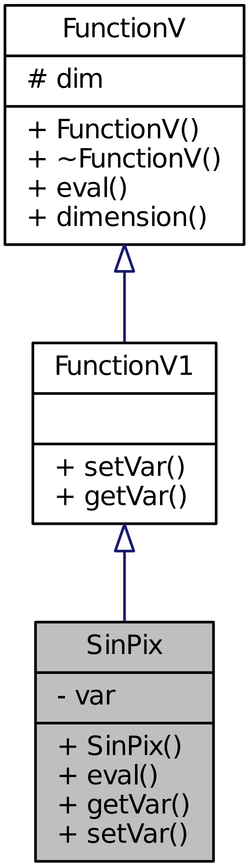
The documentation for this class was generated from the following file:

- SinPinxSinPinySinPinz.hpp

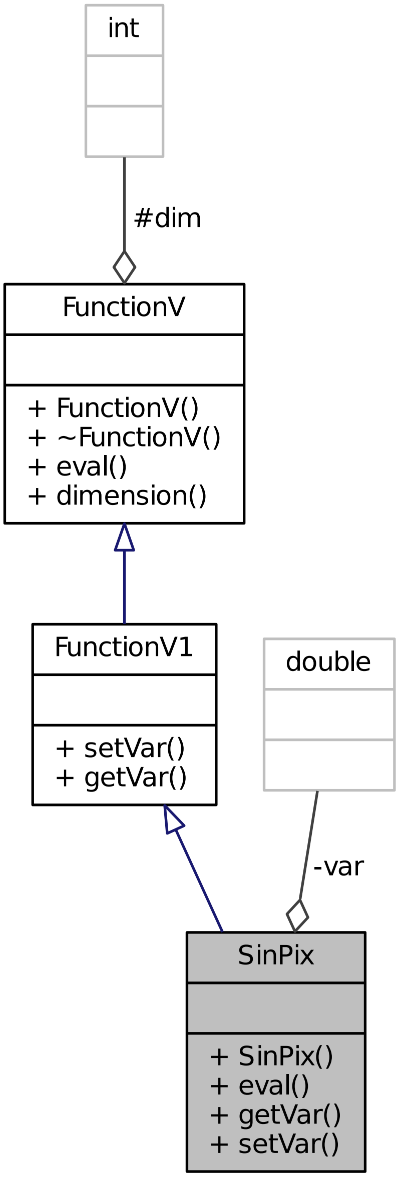
7.65 SinPix Class Reference

```
#include <SinPix.hpp>
```

Inheritance diagram for SinPix:



Collaboration diagram for SinPix:



Public Member Functions

- SinPix (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.65.1 Constructor & Destructor Documentation

7.65.1.1 SinPix::SinPix (Idouble *b*) [inline]

7.65.2 Member Function Documentation

7.65.2.1 Idouble SinPix::eval (int *d*, Idouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.65.2.2 Idouble SinPix::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.65.2.3 void SinPix::setVar (Idouble *b*) [inline], [virtual]

Implements FunctionV1.

7.65.3 Member Data Documentation

7.65.3.1 Idouble SinPix::var [private]

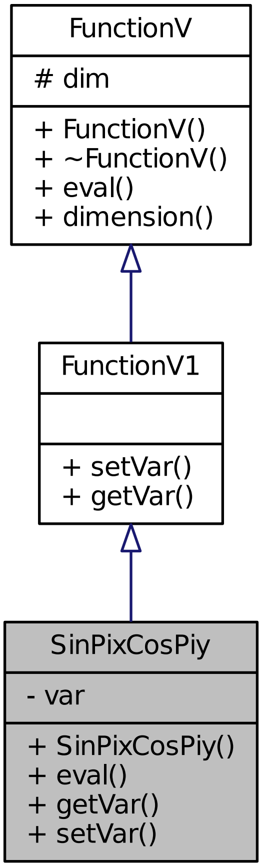
The documentation for this class was generated from the following file:

- SinPix.hpp

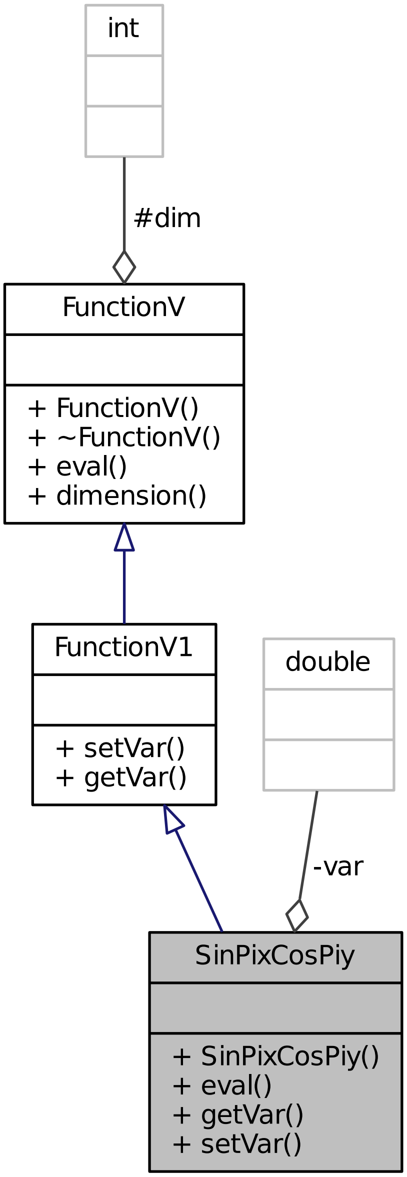
7.66 SinPixCosPiy Class Reference

```
#include <SinPixCosPiy.hpp>
```

Inheritance diagram for SinPixCosPiy:



Collaboration diagram for SinPixCosPiy:



Public Member Functions

- SinPixCosPiy (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.66.1 Constructor & Destructor Documentation

7.66.1.1 SinPixCosPiy::SinPixCosPiy (Idouble *b*) [inline]

7.66.2 Member Function Documentation

7.66.2.1 Idouble SinPixCosPiy::eval (int *d,* Idouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.66.2.2 Idouble SinPixCosPiy::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.66.2.3 void SinPixCosPiy::setVar (Idouble *b*) [inline], [virtual]

Implements FunctionV1.

7.66.3 Member Data Documentation

7.66.3.1 Idouble SinPixCosPiy::var [private]

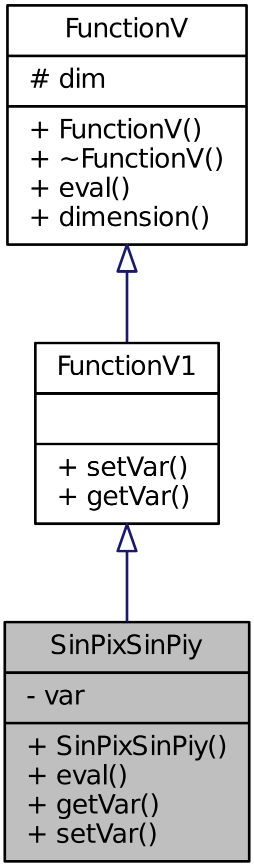
The documentation for this class was generated from the following file:

- SinPixCosPiy.hpp

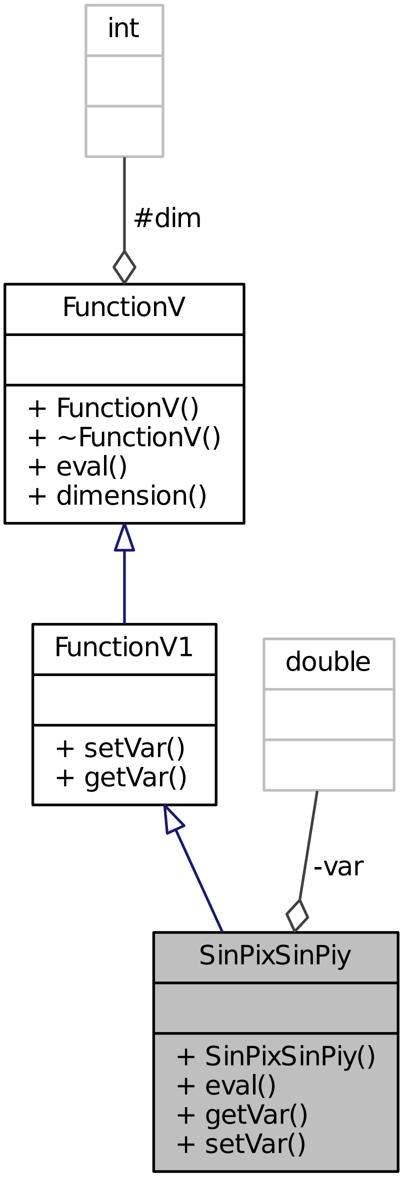
7.67 SinPixSinPiy Class Reference

```
#include <SinPixSinPiy.hpp>
```

Inheritance diagram for SinPixSinPiy:



Collaboration diagram for SinPixSinPiy:



Public Member Functions

- SinPixSinPiy (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.67.1 Constructor & Destructor Documentation

7.67.1.1 SinPixSinPiy::SinPixSinPiy (Idouble *b*) [inline]

7.67.2 Member Function Documentation

7.67.2.1 Idouble SinPixSinPiy::eval (int *d,* Idouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.67.2.2 Idouble SinPixSinPiy::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.67.2.3 void SinPixSinPiy::setVar (Idouble *b*) [inline], [virtual]

Implements FunctionV1.

7.67.3 Member Data Documentation

7.67.3.1 Idouble SinPixSinPiy::var [private]

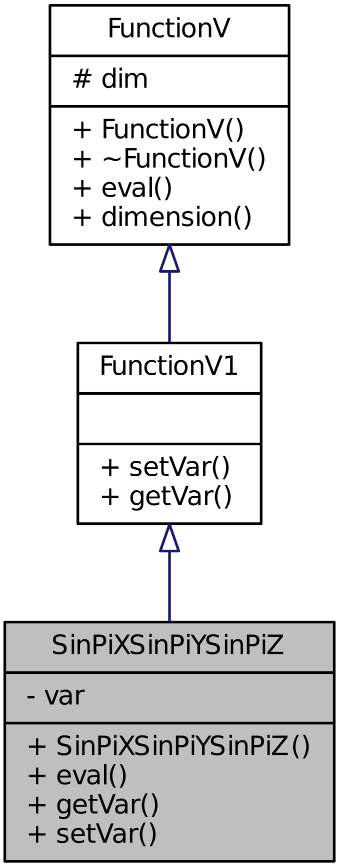
The documentation for this class was generated from the following file:

- SinPixSinPiy.hpp

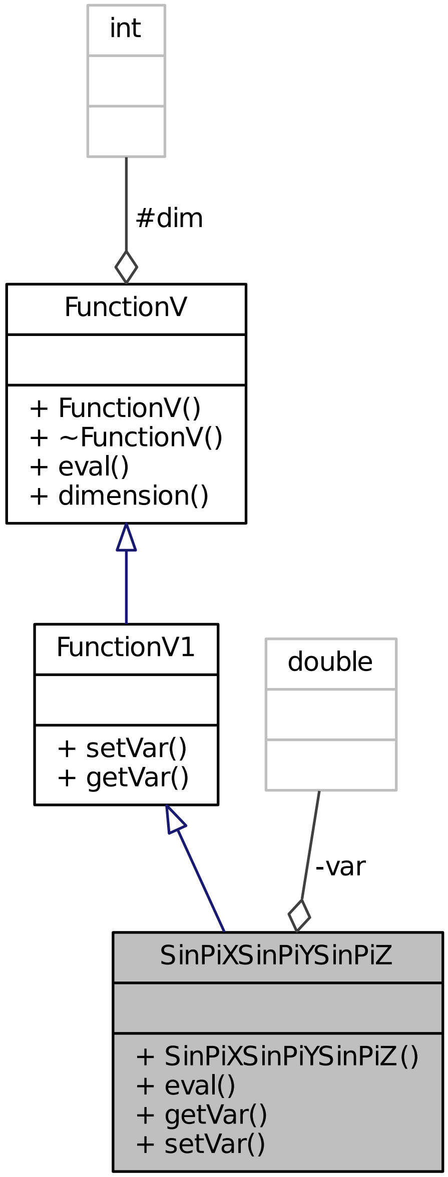
7.68 SinPiXSinPiYSinPiZ Class Reference

```
#include <SinPiXSinPiYSinPiZ.hpp>
```

Inheritance diagram for SinPiXSinPiYSinPiZ:



Collaboration diagram for SinPiXSinPiYSinPiZ:



Public Member Functions

- SinPiXSinPiYSinPiZ (ldouble b)
- ldouble eval (int d, ldouble ∗x)
- ldouble getVar (void)
- void setVar (ldouble b)

Private Attributes

- Idouble var

Additional Inherited Members

7.68.1 Constructor & Destructor Documentation

7.68.1.1 SinPiXSinPiYSinPiZ::SinPiXSinPiYSinPiZ (Idouble *b*) [inline]

7.68.2 Member Function Documentation

7.68.2.1 Idouble SinPiXSinPiYSinPiZ::eval (int *d*, Idouble ∗ *x*) [inline], [virtual]

Implements FunctionV.

7.68.2.2 Idouble SinPiXSinPiYSinPiZ::getVar (void) [inline], [virtual]

Implements FunctionV1.

7.68.2.3 void SinPiXSinPiYSinPiZ::setVar (Idouble *b*) [inline], [virtual]

Implements FunctionV1.

7.68.3 Member Data Documentation

7.68.3.1 Idouble SinPiXSinPiYSinPiZ::var [private]

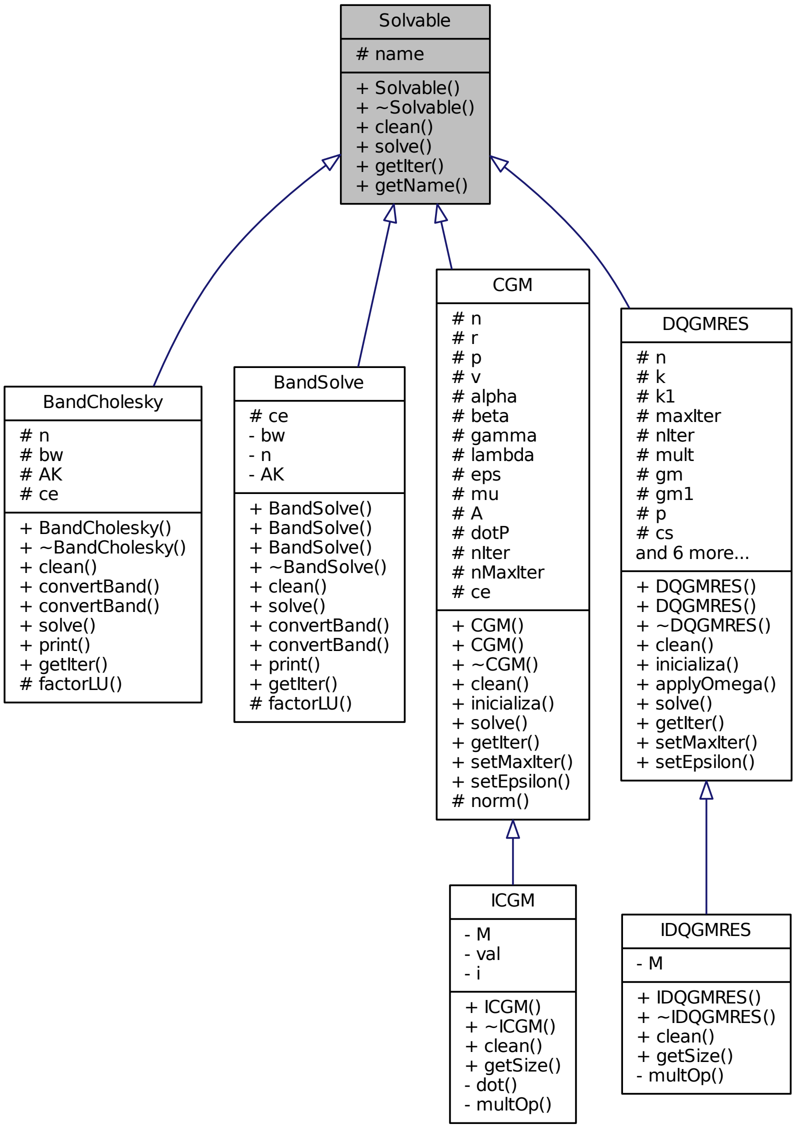
The documentation for this class was generated from the following file:

- SinPiXSinPiYSinPiZ.hpp

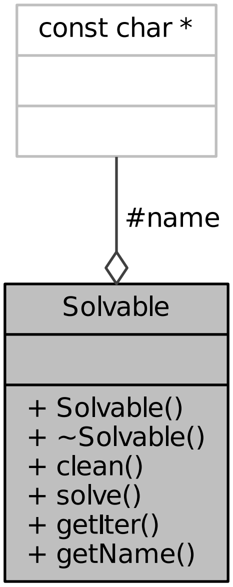
7.69 Solvable Class Reference

```
#include <Solvable.hpp>
```

Inheritance diagram for Solvable:



Collaboration diagram for Solvable:



Public Member Functions

- Solvable (void)
- virtual ∼Solvable (void)
- virtual void clean (void)=0
- virtual void solve (ldouble ∗x, ldouble ∗y)=0
- virtual int getIter (void)=0
- const char ∗ getName (void)

Protected Attributes

- const char ∗ name

7.69.1 Constructor & Destructor Documentation

7.69.1.1 Solvable::Solvable (void) `[inline]`

7.69.1.2 virtual Solvable::∼Solvable (void) `[inline], [virtual]`

7.69.2 Member Function Documentation

7.69.2.1 virtual void Solvable::clean (void) [pure virtual]

Implemented in CGM, DQGMRES, ICGM, BandSolve, IDQGMRES, and BandCholesky.

7.69.2.2 virtual int Solvable::getIter (void) [pure virtual]

Implemented in DQGMRES, CGM, BandCholesky, and BandSolve.

7.69.2.3 const char∗ Solvable::getName (void) [inline]

7.69.2.4 virtual void Solvable::solve (Idouble ∗ *x*, Idouble ∗ *y*) [pure virtual]

Implemented in DQGMRES, CGM, BandCholesky, and BandSolve.

7.69.3 Member Data Documentation

7.69.3.1 const char∗ Solvable::name [protected]

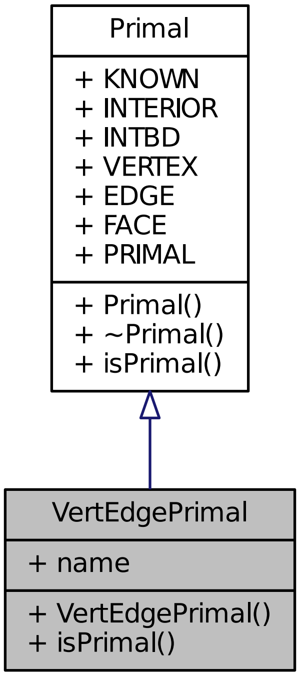
The documentation for this class was generated from the following file:

- Solvable.hpp

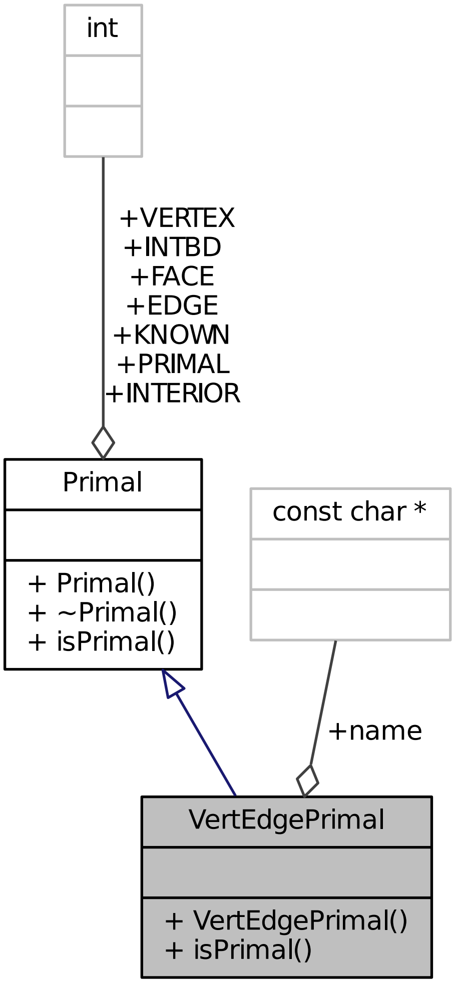
7.70 VertEdgePrimal Class Reference

```
#include <VertEdgePrimal.hpp>
```

Inheritance diagram for VertEdgePrimal:



Collaboration diagram for VertEdgePrimal:



Public Member Functions

- VertEdgePrimal (void)
- bool isPrimal (int type, int ∗coordN, int ∗coordM)

Public Attributes

- const char ∗ name

Additional Inherited Members

7.70.1 Constructor & Destructor Documentation

7.70.1.1 VertEdgePrimal::VertEdgePrimal (void) [inline]

7.70.2 Member Function Documentation

7.70.2.1 bool VertEdgePrimal::isPrimal (int *type,* int ∗ *coordN,* int ∗ *coordM*) [inline], [virtual]

Implements Primal.

7.70.3 Member Data Documentation

7.70.3.1 const char∗ VertEdgePrimal::name

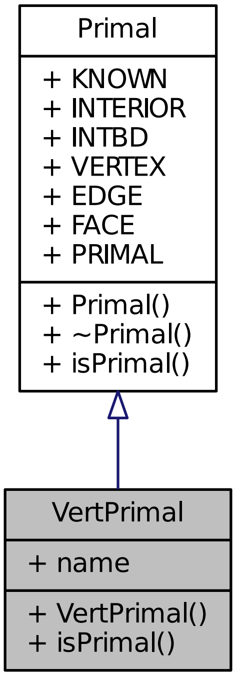
The documentation for this class was generated from the following file:

- VertEdgePrimal.hpp

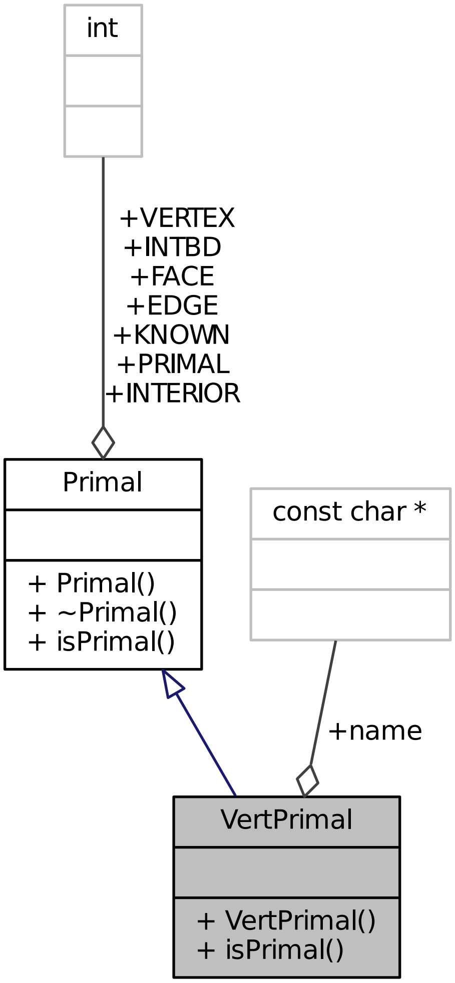
7.71 VertPrimal Class Reference

```
#include <VertPrimal.hpp>
```

Inheritance diagram for VertPrimal:



Collaboration diagram for VertPrimal:



Public Member Functions

- VertPrimal (void)
- bool isPrimal (int type, int ∗coordN, int ∗coordM)

Public Attributes

- const char ∗ name

Additional Inherited Members

7.71.1 Constructor & Destructor Documentation

7.71.1.1 VertPrimal::VertPrimal (void) [inline]

7.71.2 Member Function Documentation

7.71.2.1 bool VertPrimal::isPrimal (int *type,* int ∗ *coordN,* int ∗ *coordM*) [inline], [virtual]

Implements Primal.

7.71.3 Member Data Documentation

7.71.3.1 const char∗ VertPrimal::name

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

- VertPrimal.hpp

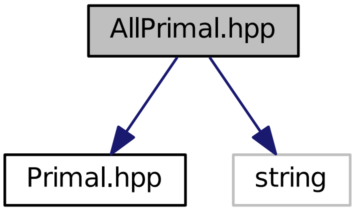
Chapter 8

File Documentation

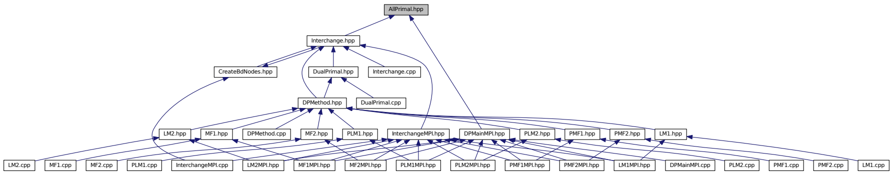
8.1 AllPrimal.hpp File Reference

```
#include "Primal.hpp"
#include <string>
```

Include dependency graph for AllPrimal.hpp:



This graph shows which files directly or indirectly include this file:



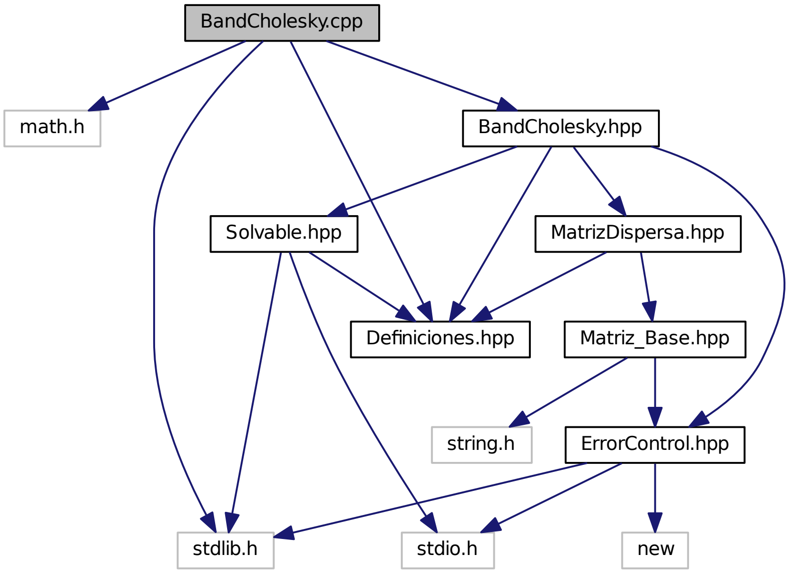
Classes

- class AllPrimal

8.2 BandCholesky.cpp File Reference

```
#include <math.h>
#include <stdlib.h>
#include "Definiciones.hpp"
#include "BandCholesky.hpp"
```

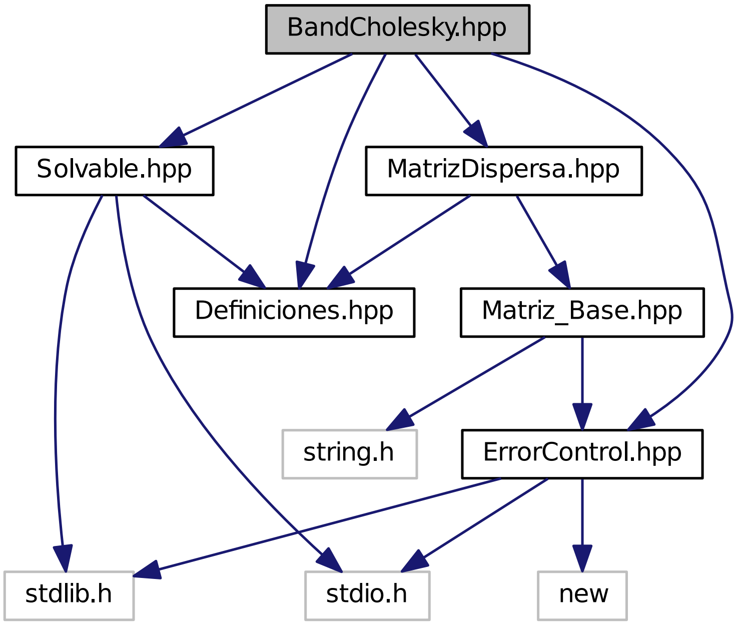
Include dependency graph for BandCholesky.cpp:



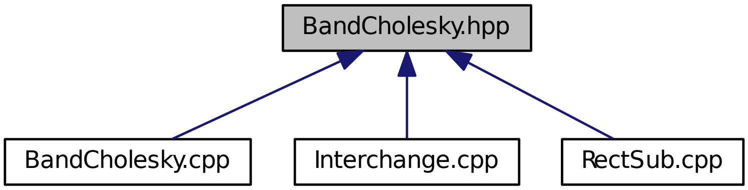
8.3 BandCholesky.hpp File Reference

```
#include "Definiciones.hpp"
#include "Solvable.hpp"
#include "MatrizDispersa.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for BandCholesky.hpp:



This graph shows which files directly or indirectly include this file:



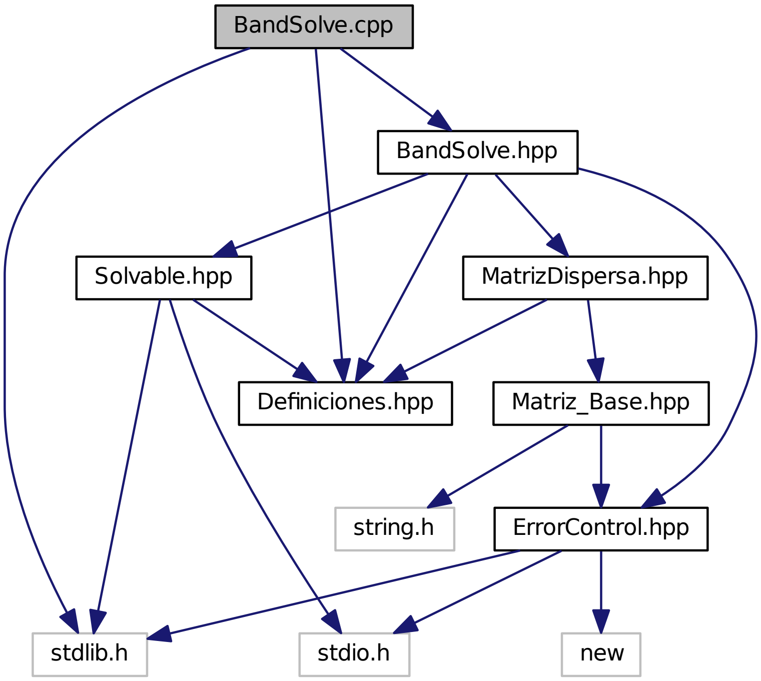
Classes

- class BandCholesky

8.4 BandSolve.cpp File Reference

```
#include <stdlib.h>
#include "Definiciones.hpp"
#include "BandSolve.hpp"
```

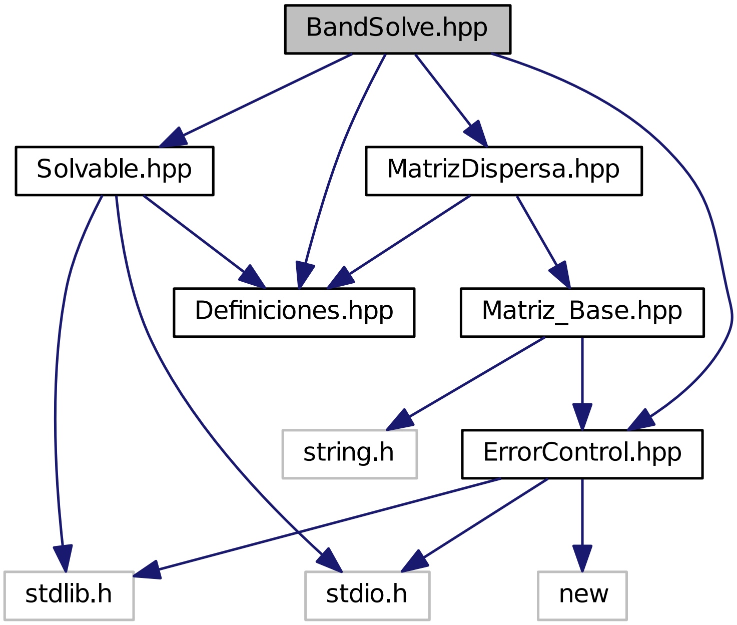
Include dependency graph for BandSolve.cpp:



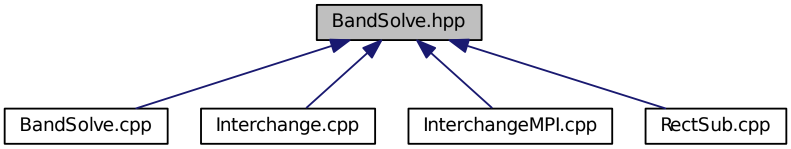
8.5 BandSolve.hpp File Reference

```
#include "Definiciones.hpp"
#include "Solvable.hpp"
#include "MatrizDispersa.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for BandSolve.hpp:



This graph shows which files directly or indirectly include this file:

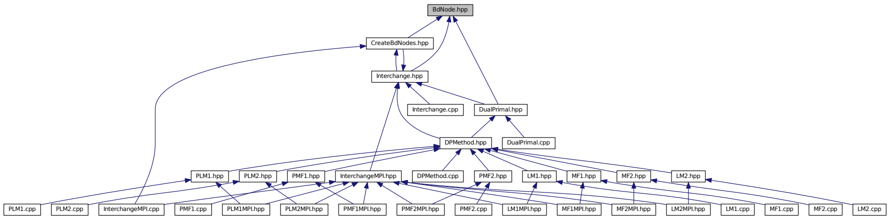


Classes

- class BandSolve

8.6 BdNode.hpp File Reference

This graph shows which files directly or indirectly include this file:



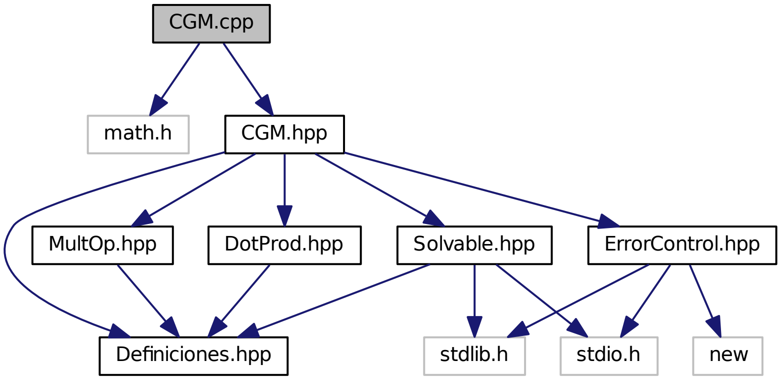
Classes

- class BdNode

8.7 CGM.cpp File Reference

```
#include <math.h>
#include "CGM.hpp"
```

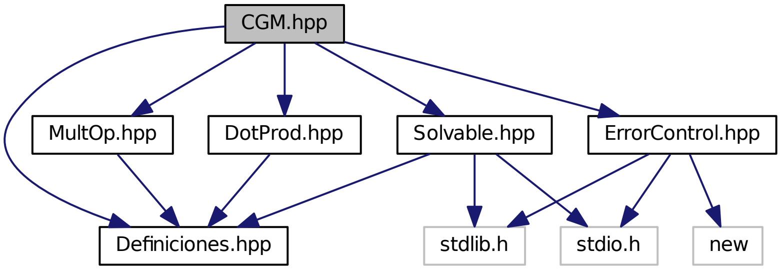
Include dependency graph for CGM.cpp:



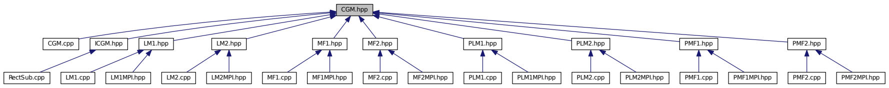
8.8 CGM.hpp File Reference

```
#include "Definiciones.hpp"
#include "Solvable.hpp"
#include "MultOp.hpp"
#include "DotProd.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for CGM.hpp:



This graph shows which files directly or indirectly include this file:



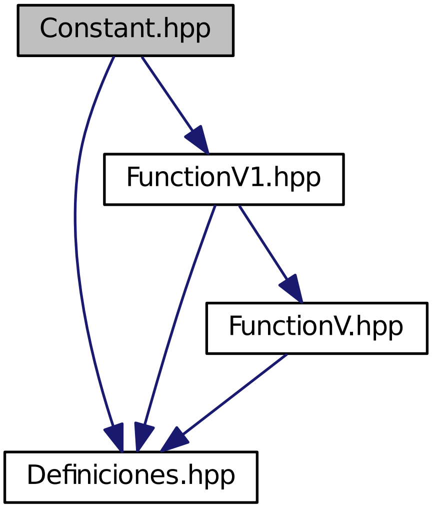
Classes

- class CGM

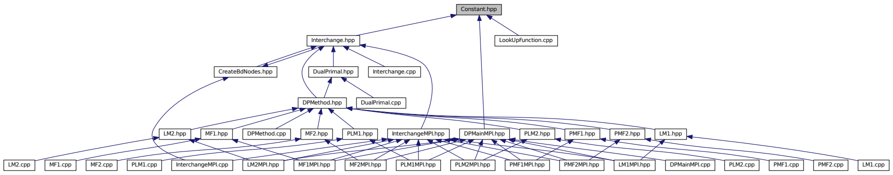
8.9 Constant.hpp File Reference

```
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for Constant.hpp:



This graph shows which files directly or indirectly include this file:



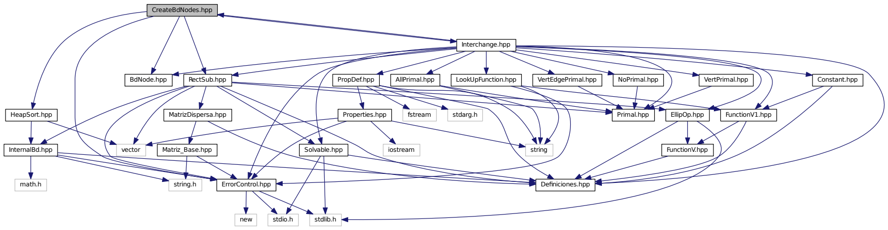
Classes

- class Constant

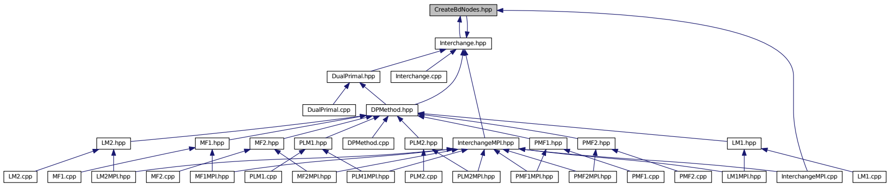
8.10 CreateBdNodes.hpp File Reference

```
#include "RectSub.hpp"
#include "BdNode.hpp"
#include "HeapSort.hpp"
#include "Interchange.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for CreateBdNodes.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class CreateBdNodes

8.11 Definiciones.hpp File Reference

Macros

- #define NMAXITER 5000

  Usa la libreria libgmm++-dev.
- #define NMAXITER_LOCAL 50000
- #define EPSILON 1e-6

  Tolerancia en los metodos iterativos.
- #define EPSILON_LOCAL (EPSILON/1e+4)
- #define EPS_EQUAL 1e-15

  Se toman como iguales dos nodos que difieran en menos que esta EPS_EQUAL.
- #define DIM_VECTOR 1

  Con esta opcion visualiza o no el residual de cada iteracion.
- #define COEFICIENTES_CONSTANTES

  Con esta opcion se calcula el numero de condicionamiento en los metodos precondicionados.
- #define __Double__

  Activada para trabajar con numeros double en caso contrario trabajar con long double.

Typedefs

- typedef double ldouble

  Define ldouble como double.

8.11.1 Macro Definition Documentation

8.11.1.1 #define __Double__

Activada para trabajar con numeros double en caso contrario trabajar con long double.

8.11.1.2 #define COEFICIENTES_CONSTANTES

Con esta opcion se calcula el numero de condicionamiento en los metodos precondicionados.

Activar el modo de depuracion Definiciones Generales, en caso de no existir definicion generales, solo se consideran coeficientes constantes Definicion de problemas que requieren activar codigo particular para cada problema de ejemplo Activacion de las diferentes definiciones para cada problema

8.11.1.3 #define DIM_VECTOR 1

Con esta opcion visualiza o no el residual de cada iteracion.

Dimension del vector (1) escalar

8.11.1.4 #define EPS_EQUAL 1e-15

Se toman como iguales dos nodos que difieran en menos que esta EPS_EQUAL.

8.11.1.5 #define EPSILON 1e-6

Tolerancia en los metodos iterativos.

8.11.1.6 #define EPSILON_LOCAL (EPSILON/1e+4)

8.11.1.7 #define NMAXITER 5000

Usa la libreria libgmm++-dev.

Numero maximo de iteraiones en los metodos iterativos

8.11.1.8 #define NMAXITER_LOCAL 50000

8.11.2 Typedef Documentation

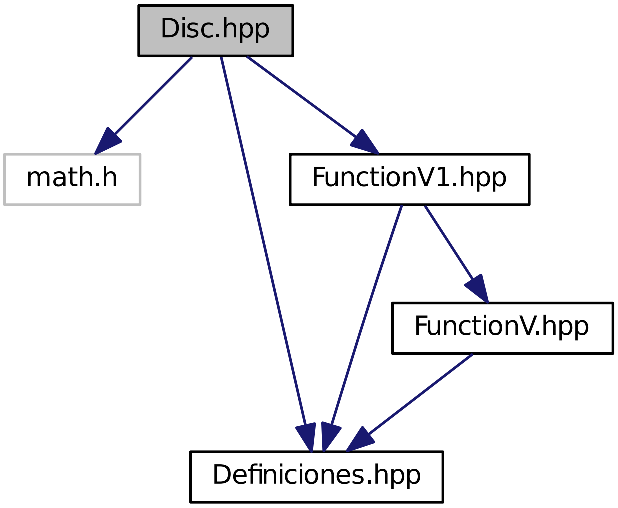
8.11.2.1 typedef double ldouble

Define ldouble como double.

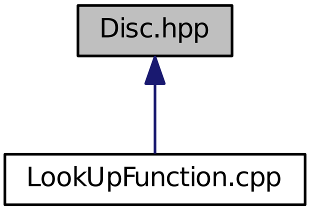
8.12 Disc.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for Disc.hpp:



This graph shows which files directly or indirectly include this file:



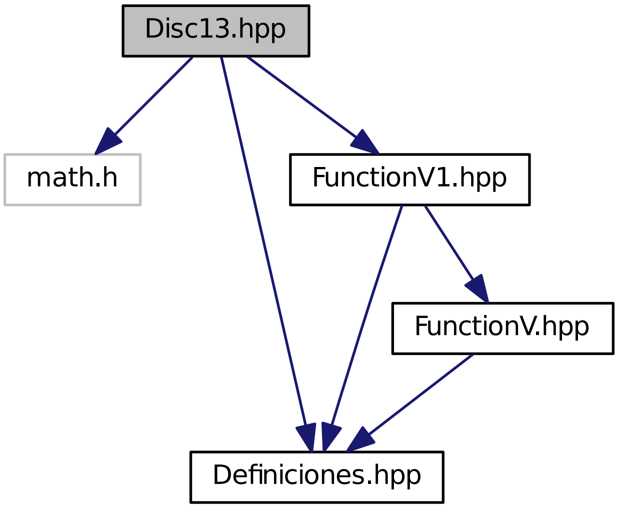
Classes

- class Disc

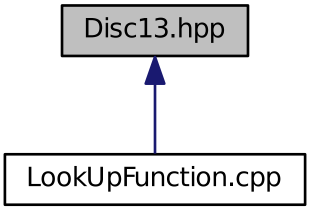
8.13 Disc13.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for Disc13.hpp:



This graph shows which files directly or indirectly include this file:



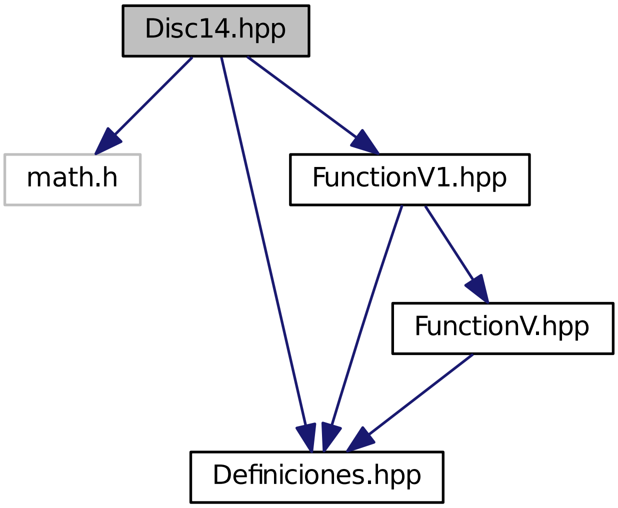
Classes

- class Disc13

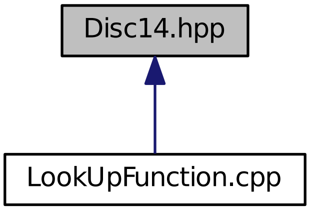
8.14 Disc14.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for Disc14.hpp:



This graph shows which files directly or indirectly include this file:



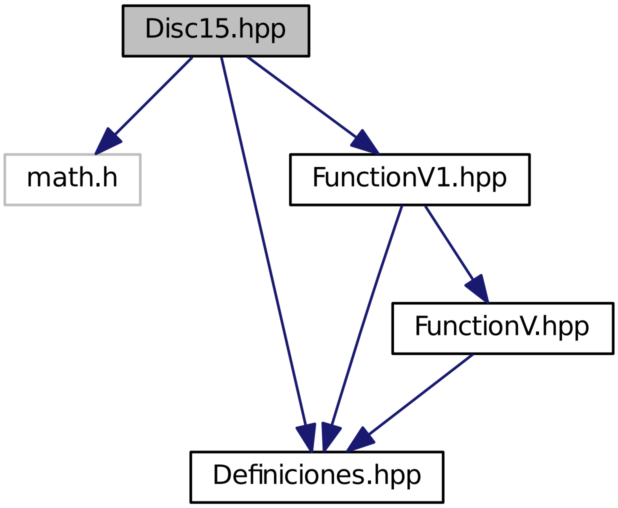
Classes

- class Disc14

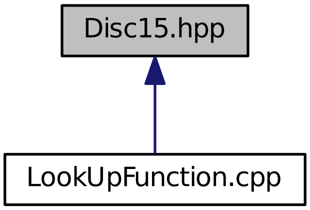
8.15 Disc15.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for Disc15.hpp:



This graph shows which files directly or indirectly include this file:



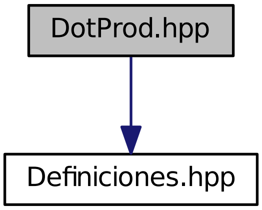
Classes

- class Disc15

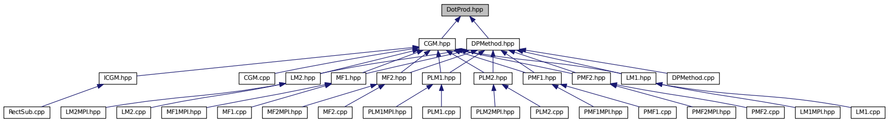
8.16 DotProd.hpp File Reference

```
#include "Definiciones.hpp"
```

Include dependency graph for DotProd.hpp:



This graph shows which files directly or indirectly include this file:



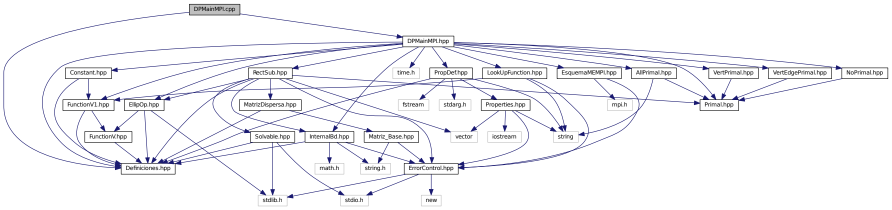
Classes

- class DotProd

8.17 DPMainMPI.cpp File Reference

```
#include "Definiciones.hpp"
#include "DPMainMPI.hpp"
```

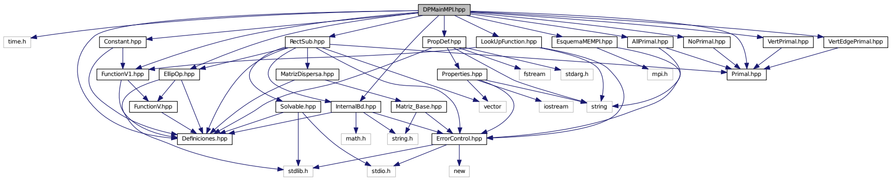
Include dependency graph for DPMainMPI.cpp:



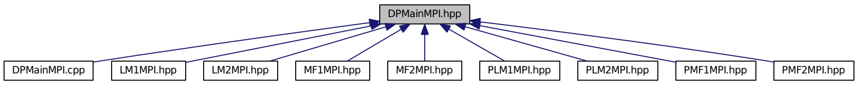
8.18 DPMainMPI.hpp File Reference

```
#include <time.h>
#include "Definiciones.hpp"
#include "EsquemaMEMPI.hpp"
#include "PropDef.hpp"
#include "EllipOp.hpp"
#include "InternalBd.hpp"
#include "FunctionV1.hpp"
#include "Primal.hpp"
#include "Constant.hpp"
#include "LookUpFunction.hpp"
#include "VertPrimal.hpp"
#include "VertEdgePrimal.hpp"
#include "AllPrimal.hpp"
#include "NoPrimal.hpp"
#include "RectSub.hpp"
```

Include dependency graph for DPMainMPI.hpp:



This graph shows which files directly or indirectly include this file:



Classes

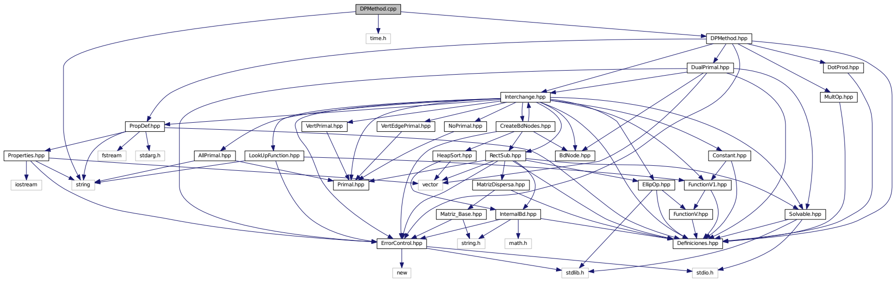
- class DPMainMPI

    Clase base para definir a los metodos DVS-DDM.

8.19 DPMethod.cpp File Reference

```
#include <string>
#include <time.h>
#include "DPMethod.hpp"
```

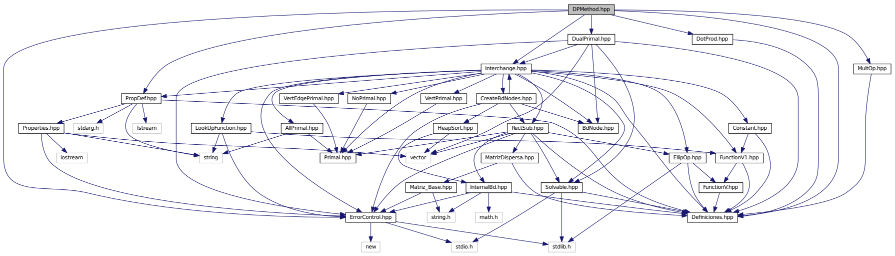
Include dependency graph for DPMethod.cpp:



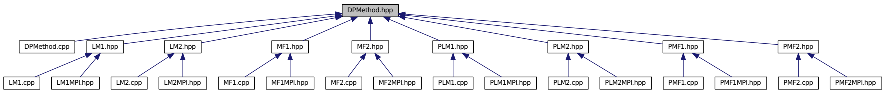
8.20 DPMethod.hpp File Reference

```
#include "Definiciones.hpp"
#include "MultOp.hpp"
#include "DotProd.hpp"
#include "PropDef.hpp"
#include "DualPrimal.hpp"
#include "Interchange.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for DPMethod.hpp:



This graph shows which files directly or indirectly include this file:



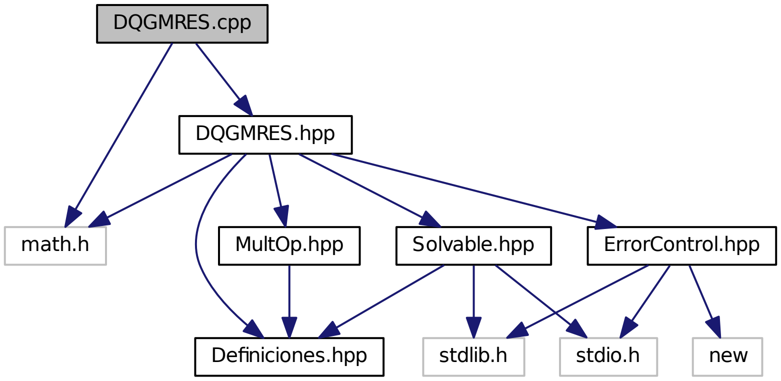
Classes

- class DPMethod

8.21 DQGMRES.cpp File Reference

```
#include <math.h>
#include "DQGMRES.hpp"
```

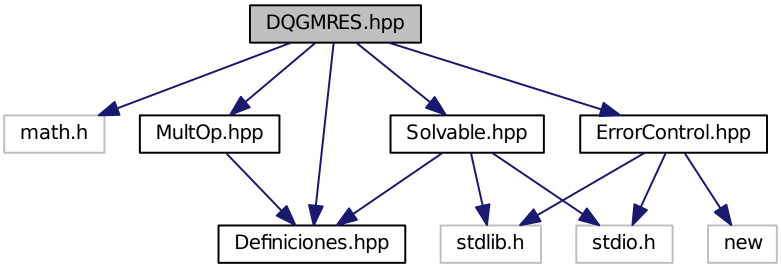
Include dependency graph for DQGMRES.cpp:



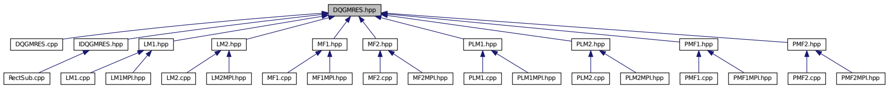
8.22 DQGMRES.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "Solvable.hpp"
#include "MultOp.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for DQGMRES.hpp:



This graph shows which files directly or indirectly include this file:



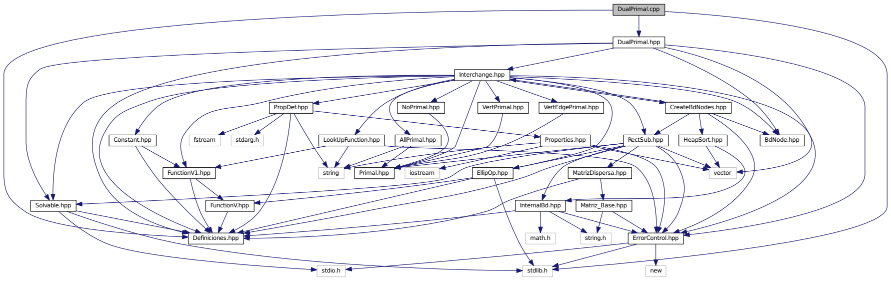
Classes

- class DQGMRES

8.23 DualPrimal.cpp File Reference

```
#include <stdlib.h>
#include "Definiciones.hpp"
#include "DualPrimal.hpp"
```

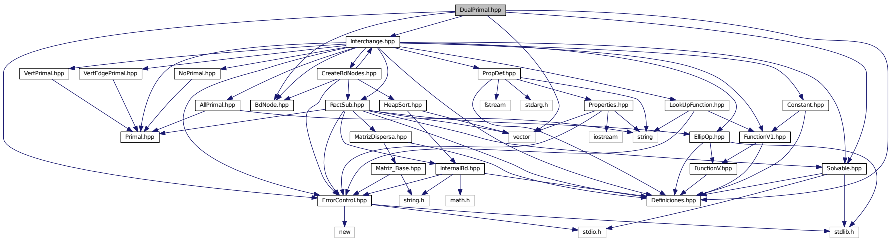
Include dependency graph for DualPrimal.cpp:



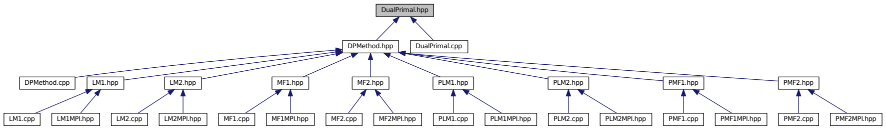
8.24 DualPrimal.hpp File Reference

```
#include <vector>
#include "Definiciones.hpp"
#include "BdNode.hpp"
#include "Solvable.hpp"
#include "Interchange.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for DualPrimal.hpp:



This graph shows which files directly or indirectly include this file:



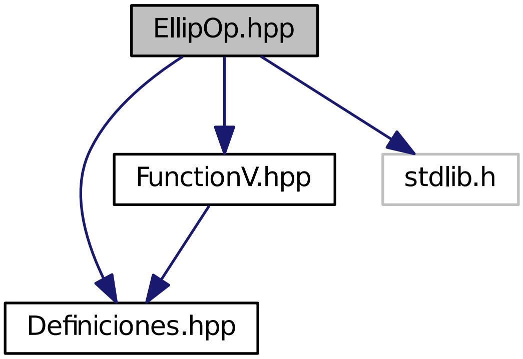
Classes

- class DualPrimal

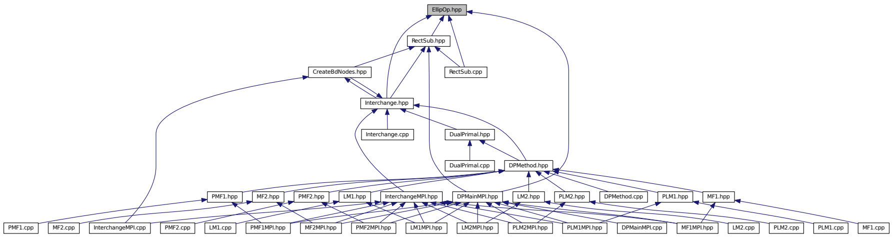
8.25 EllipOp.hpp File Reference

```
#include "Definiciones.hpp"
#include "FunctionV.hpp"
#include <stdlib.h>
```

Include dependency graph for EllipOp.hpp:



This graph shows which files directly or indirectly include this file:



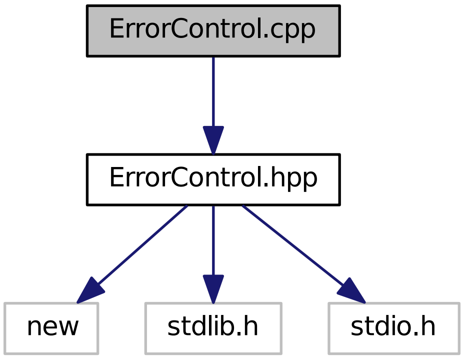
Classes

- class EllipOp

8.26 ErrorControl.cpp File Reference

```
#include "ErrorControl.hpp"
```

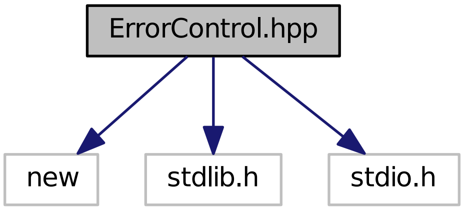
Include dependency graph for ErrorControl.cpp:



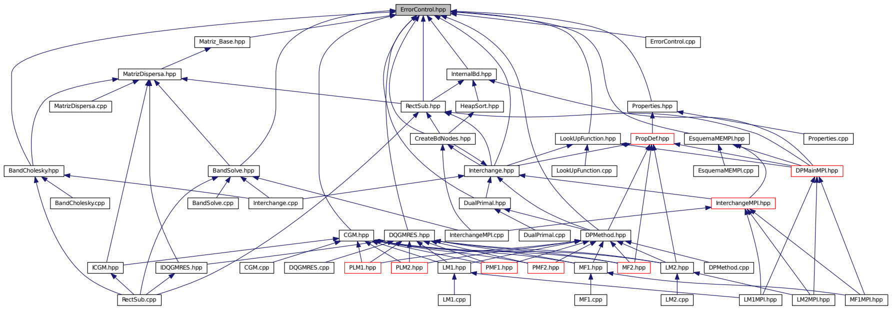
8.27 ErrorControl.hpp File Reference

```
#include <new>
#include <stdlib.h>
#include <stdio.h>
```

Include dependency graph for ErrorControl.hpp:



This graph shows which files directly or indirectly include this file:



Classes

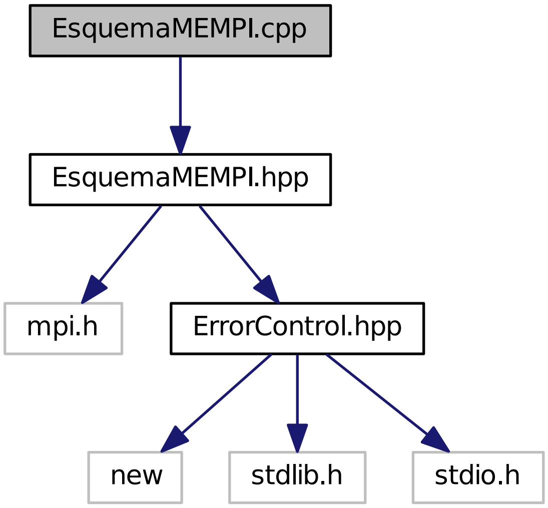
- class ErrorControl

  Error Control.

8.28 EsquemaMEMPI.cpp File Reference

```
#include "EsquemaMEMPI.hpp"
```

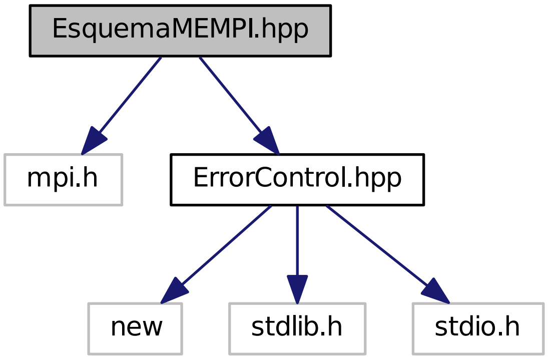
Include dependency graph for EsquemaMEMPI.cpp:



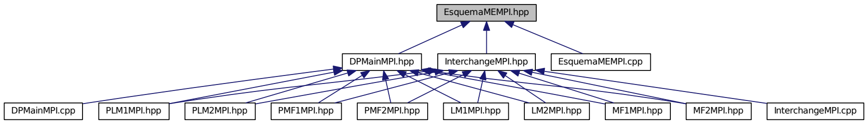
8.29 EsquemaMEMPI.hpp File Reference

```
#include "mpi.h"
#include "ErrorControl.hpp"
```

Include dependency graph for EsquemaMEMPI.hpp:



This graph shows which files directly or indirectly include this file:



Classes

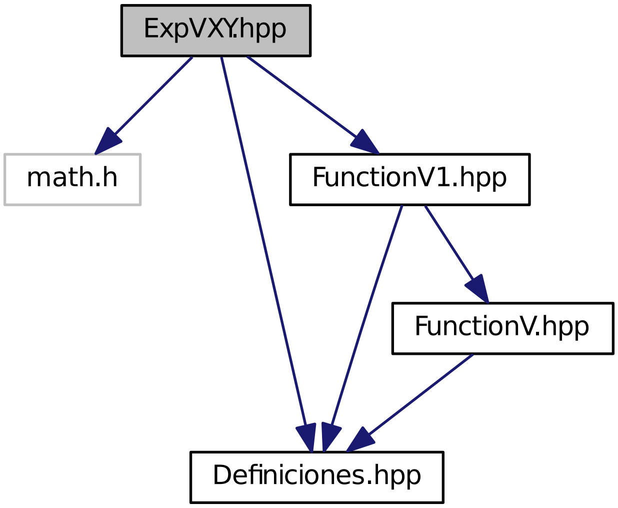
- class EsquemaMEMPI

  Clase base para definir el Esquema Maestro-Esclavo en MPI.

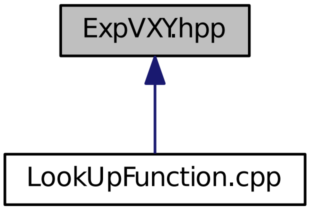
8.30 ExpVXY.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for ExpVXY.hpp:



This graph shows which files directly or indirectly include this file:



Classes

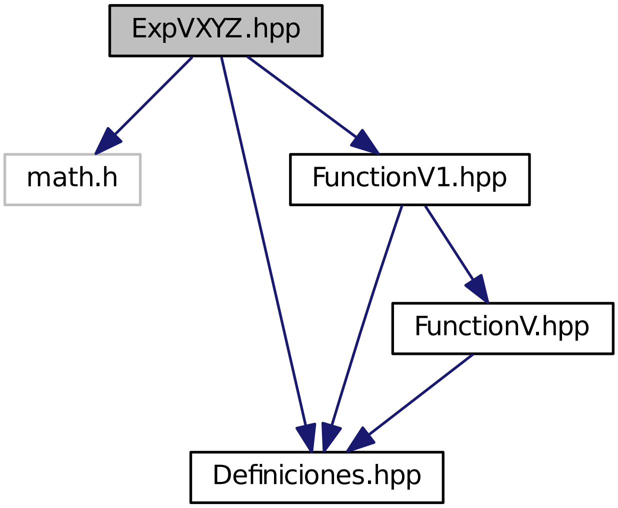
- class ExpVXY

8.31 ExpVXYZ.hpp File Reference

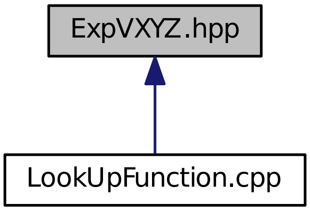
```
#include <math.h>
```

```
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for ExpVXYZ.hpp:



This graph shows which files directly or indirectly include this file:



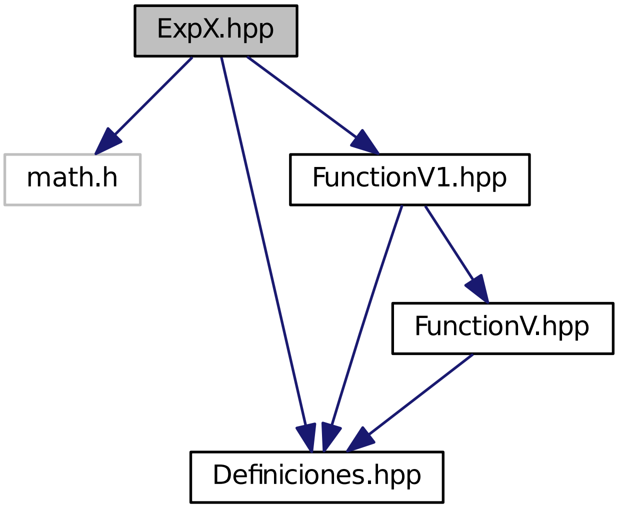
Classes

- class ExpVXYZ

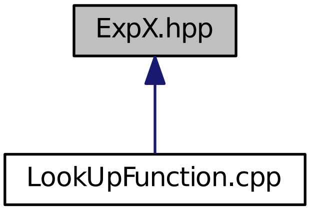
8.32 ExpX.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for ExpX.hpp:



This graph shows which files directly or indirectly include this file:



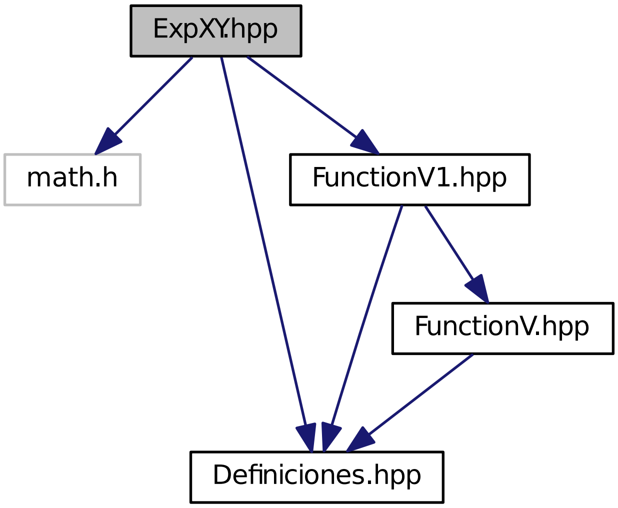
Classes

- class ExpX

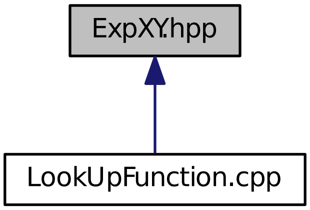
8.33 ExpXY.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for ExpXY.hpp:



This graph shows which files directly or indirectly include this file:



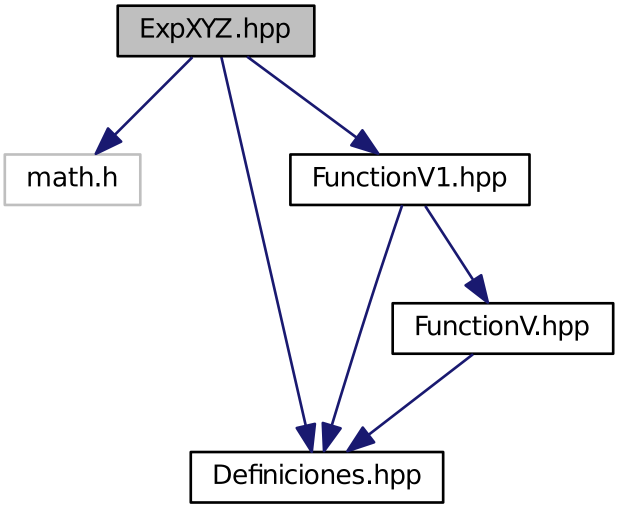
Classes

- class ExpXY

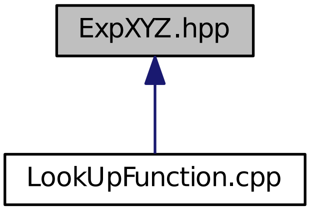
8.34 ExpXYZ.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for ExpXYZ.hpp:



This graph shows which files directly or indirectly include this file:



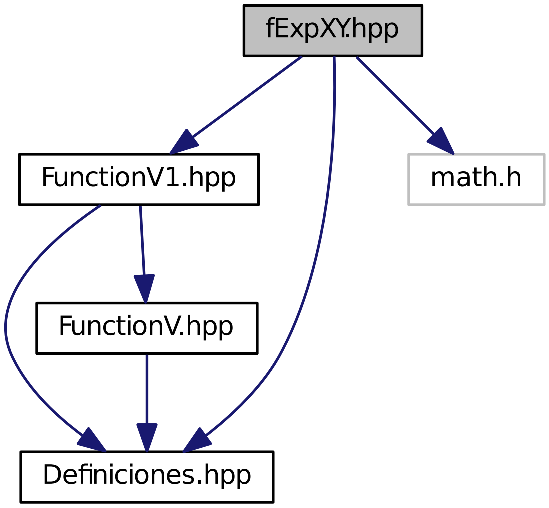
Classes

- class ExpXYZ

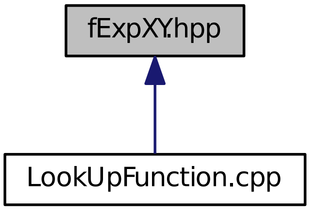
8.35 fExpXY.hpp File Reference

```
#include "FunctionV1.hpp"
#include "Definiciones.hpp"
#include <math.h>
```

Include dependency graph for fExpXY.hpp:



This graph shows which files directly or indirectly include this file:



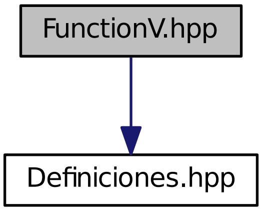
Classes

- class fExpXY

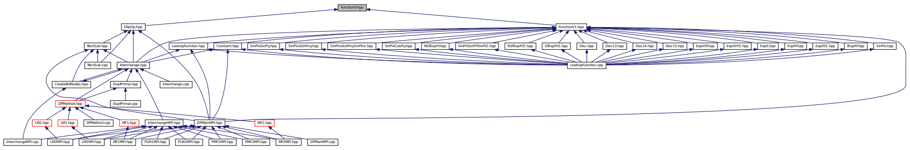
8.36 FunctionV.hpp File Reference

```
#include "Definiciones.hpp"
```

Include dependency graph for FunctionV.hpp:



This graph shows which files directly or indirectly include this file:



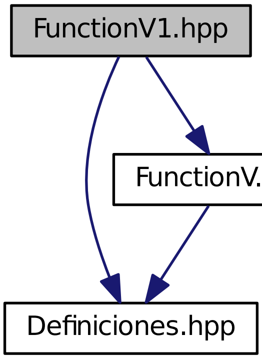
Classes

- class FunctionV

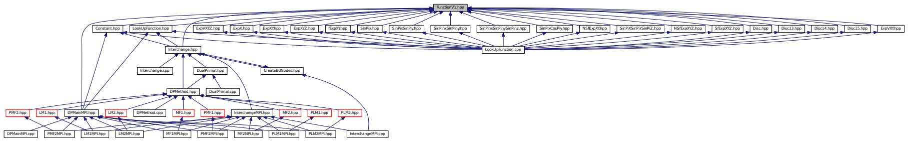
8.37 FunctionV1.hpp File Reference

```
#include "Definiciones.hpp"
#include "FunctionV.hpp"
```

Include dependency graph for FunctionV1.hpp:



This graph shows which files directly or indirectly include this file:



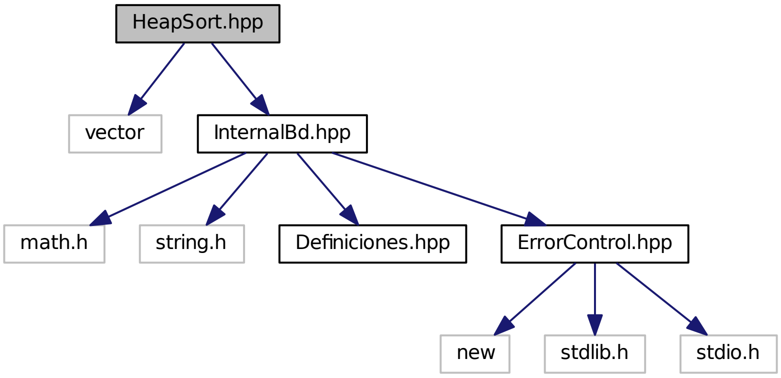
Classes

- class FunctionV1

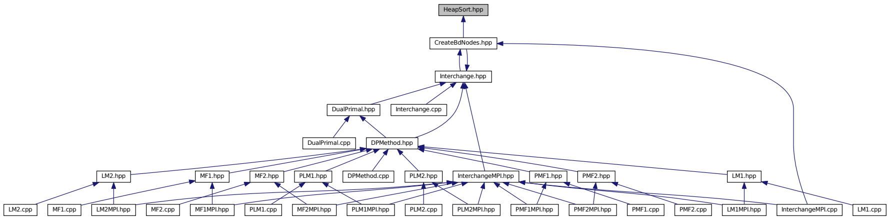
8.38 HeapSort.hpp File Reference

```
#include <vector>
#include "InternalBd.hpp"
```

Include dependency graph for HeapSort.hpp:



This graph shows which files directly or indirectly include this file:



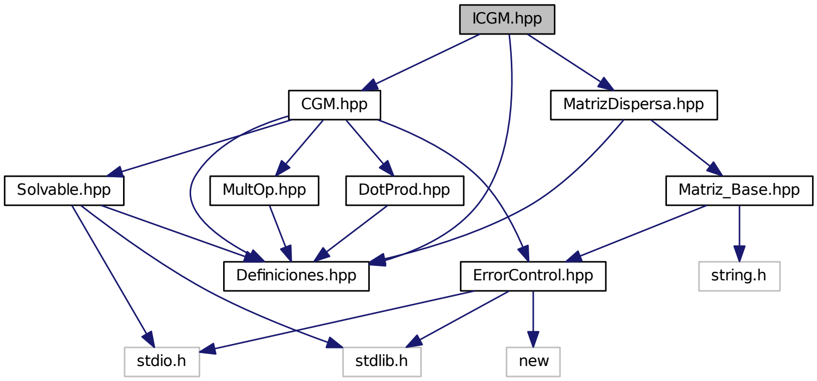
Classes

- class HeapSort

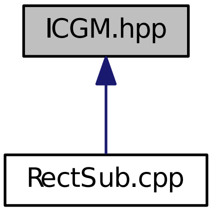
8.39 ICGM.hpp File Reference

```
#include "Definiciones.hpp"
#include "CGM.hpp"
#include "MatrizDispersa.hpp"
```

Include dependency graph for ICGM.hpp:



This graph shows which files directly or indirectly include this file:



Classes

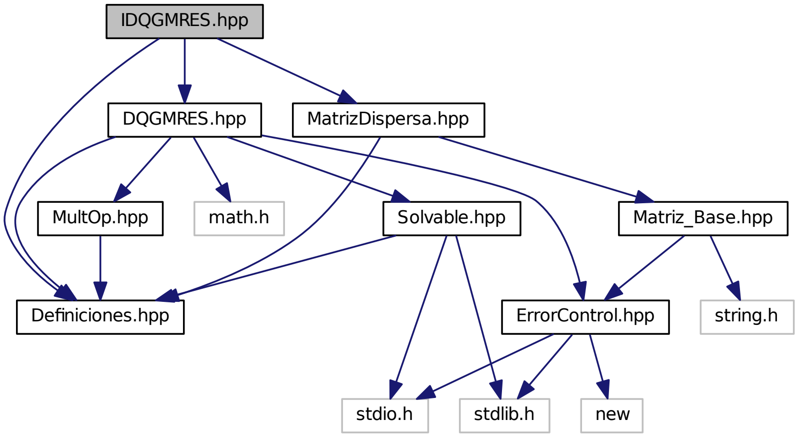
- class ICGM

  Clase para implementar CGM con matrices bandadas o dispersas.

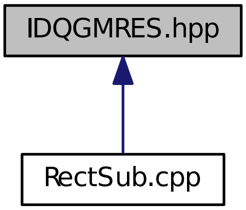
8.40 IDQGMRES.hpp File Reference

```
#include "Definiciones.hpp"
#include "DQGMRES.hpp"
#include "MatrizDispersa.hpp"
```

Include dependency graph for IDQGMRES.hpp:



This graph shows which files directly or indirectly include this file:



Classes

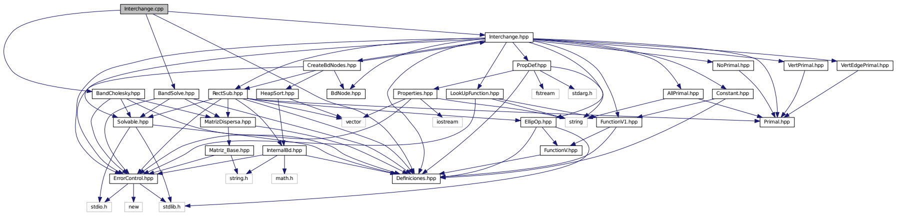
- class IDQGMRES

  Clase para implementar DQGMRES con matrices bandadas o dispersas.

8.41 Interchange.cpp File Reference

```
#include "Definiciones.hpp"
#include "Interchange.hpp"
#include "BandSolve.hpp"
#include "BandCholesky.hpp"
```

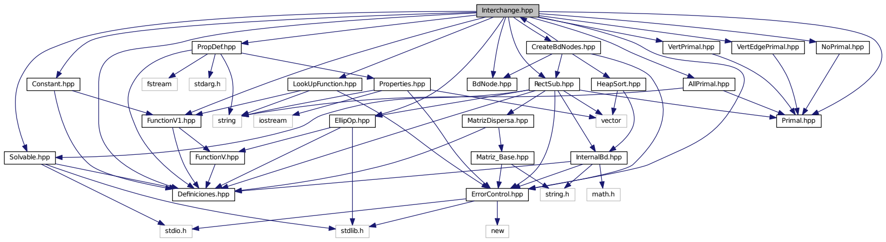
Include dependency graph for Interchange.cpp:



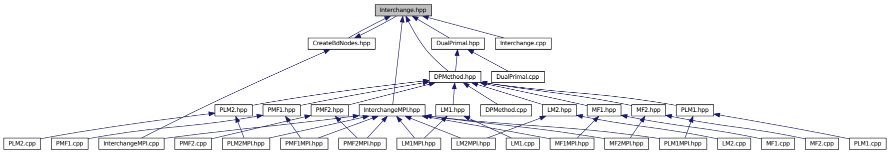
8.42 Interchange.hpp File Reference

```
#include "Definiciones.hpp"
#include "BdNode.hpp"
#include "Solvable.hpp"
#include "PropDef.hpp"
#include "RectSub.hpp"
#include "FunctionV1.hpp"
#include "Constant.hpp"
#include "EllipOp.hpp"
#include "Primal.hpp"
#include "LookUpFunction.hpp"
#include "VertPrimal.hpp"
#include "VertEdgePrimal.hpp"
#include "AllPrimal.hpp"
#include "NoPrimal.hpp"
#include "CreateBdNodes.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for Interchange.hpp:



This graph shows which files directly or indirectly include this file:



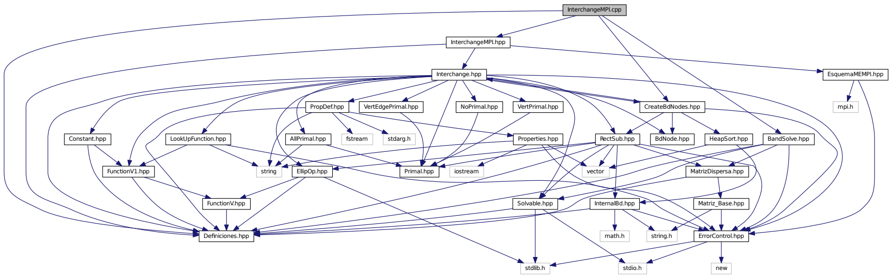
Classes

- class Interchange

8.43 InterchangeMPI.cpp File Reference

```
#include "Definiciones.hpp"
#include "InterchangeMPI.hpp"
#include "BandSolve.hpp"
#include "CreateBdNodes.hpp"
```

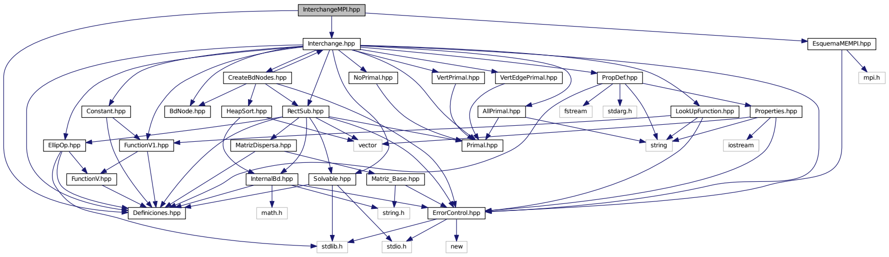
Include dependency graph for InterchangeMPI.cpp:



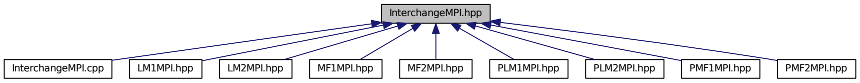
8.44 InterchangeMPI.hpp File Reference

```
#include "Definiciones.hpp"
#include "Interchange.hpp"
#include "EsquemaMEMPI.hpp"
```

Include dependency graph for InterchangeMPI.hpp:



This graph shows which files directly or indirectly include this file:



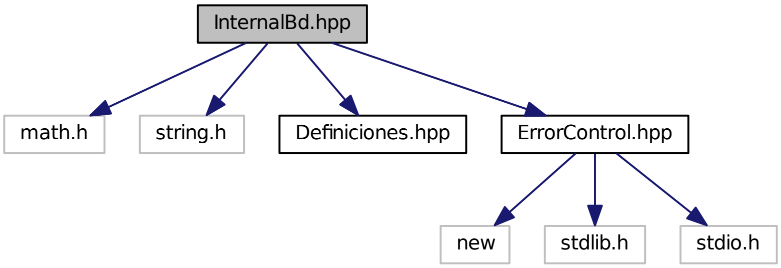
Classes

- class InterchangeMPI

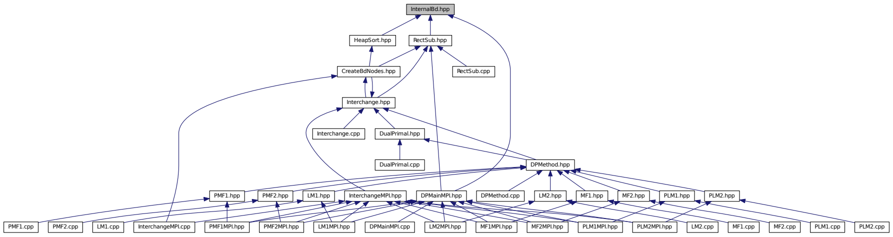
8.45 InternalBd.hpp File Reference

```
#include <math.h>
#include <string.h>
#include "Definiciones.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for InternalBd.hpp:



This graph shows which files directly or indirectly include this file:



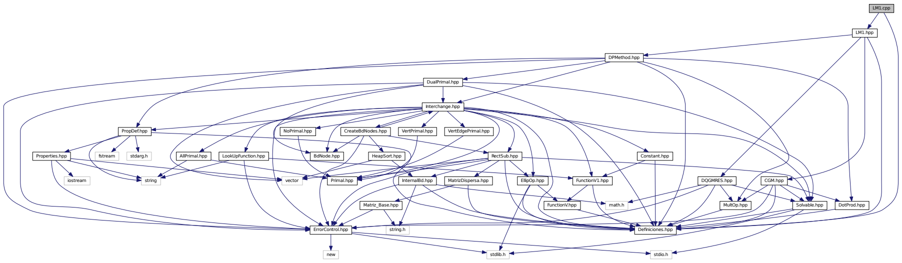
Classes

- class InternalBd

8.46 LM1.cpp File Reference

```
#include "Definiciones.hpp"
#include "LM1.hpp"
```

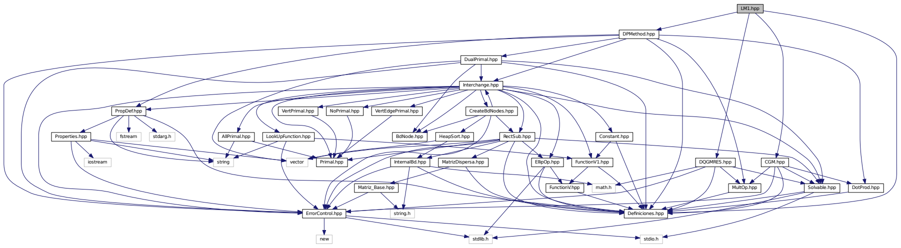
Include dependency graph for LM1.cpp:



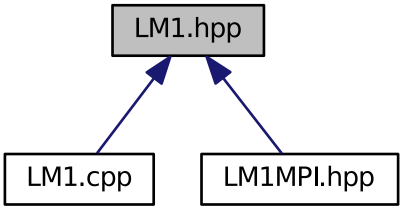
8.47 LM1.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for LM1.hpp:



This graph shows which files directly or indirectly include this file:



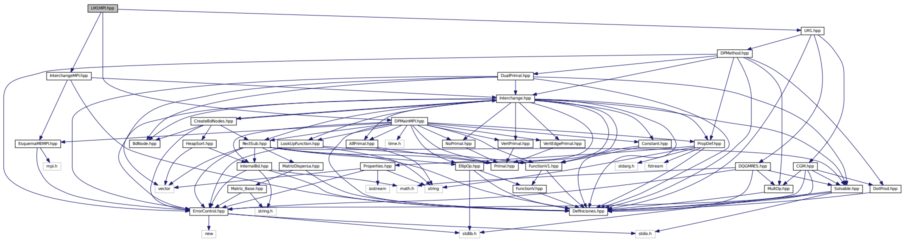
Classes

- class LM1

8.48 LM1MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "LM1.hpp"
```

Include dependency graph for LM1MPI.hpp:



Classes

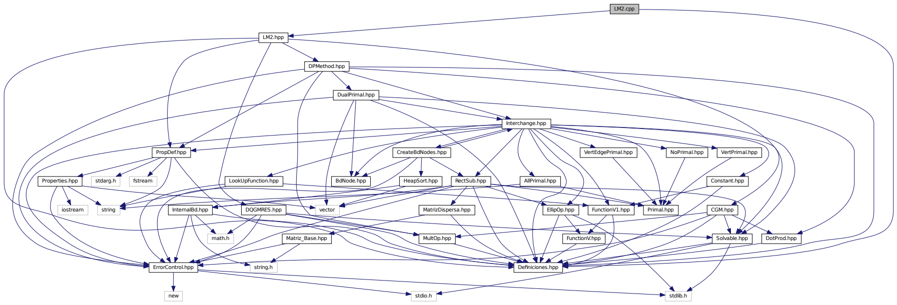
- class LM1MPI

  Clase para definir el metodo LM-1 de DVS-DDM.

8.49 LM2.cpp File Reference

```
#include "Definiciones.hpp"
#include "LM2.hpp"
```

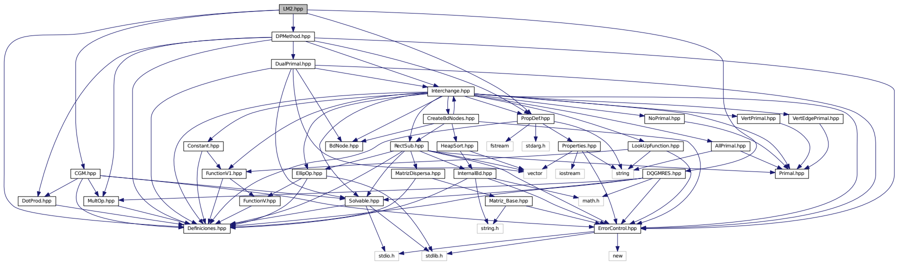
Include dependency graph for LM2.cpp:



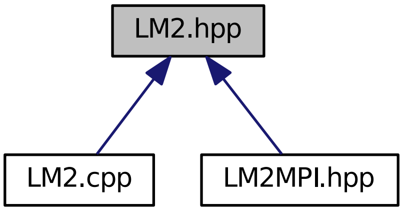
8.50 LM2.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "PropDef.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for LM2.hpp:



This graph shows which files directly or indirectly include this file:



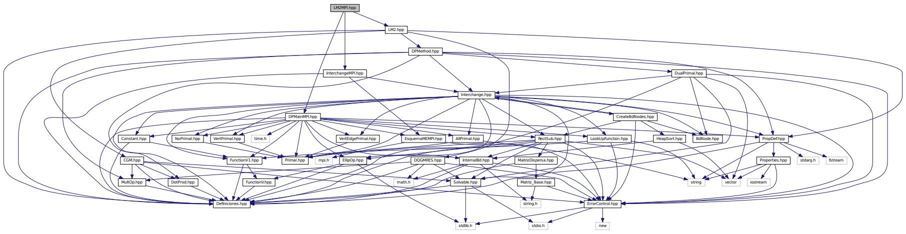
Classes

- class LM2

8.51 LM2MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "LM2.hpp"
```

Include dependency graph for LM2MPI.hpp:



Classes

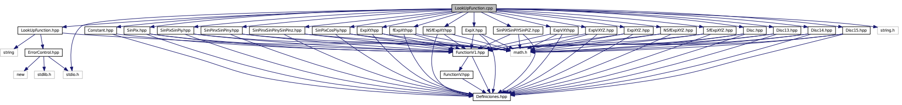
- class LM2MPI

    Clase para definir el metodo LM-2 de DVS-DDM.

8.52 LookUpFunction.cpp File Reference

```
#include "LookUpFunction.hpp"
#include "Constant.hpp"
#include "SinPix.hpp"
#include "SinPixSinPiy.hpp"
#include "SinPinxSinPiny.hpp"
#include "SinPinxSinPinySinPinz.hpp"
#include "SinPixCosPiy.hpp"
#include "ExpXY.hpp"
#include "fExpXY.hpp"
#include "NSfExpXY.hpp"
#include "ExpX.hpp"
#include "SinPiXSinPiYSinPiZ.hpp"
#include "ExpVXY.hpp"
#include "ExpVXYZ.hpp"
#include "ExpXYZ.hpp"
#include "NSfExpXYZ.hpp"
#include "SfExpXYZ.hpp"
#include "Disc.hpp"
#include "Disc13.hpp"
#include "Disc14.hpp"
#include "Disc15.hpp"
#include <string.h>
#include <stdio.h>
```

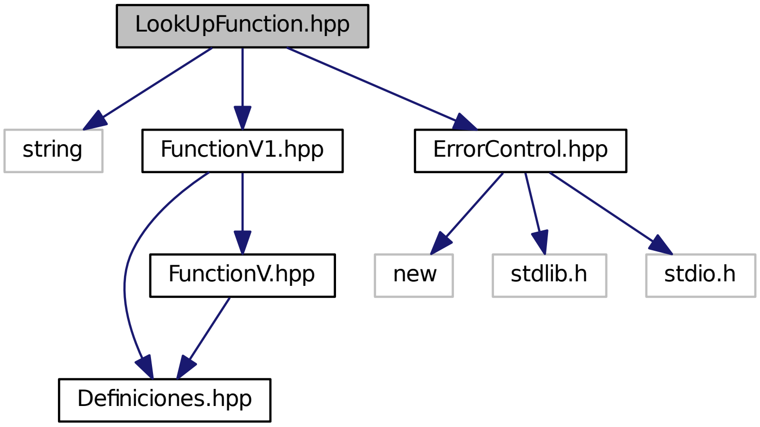
Include dependency graph for LookUpFunction.cpp:



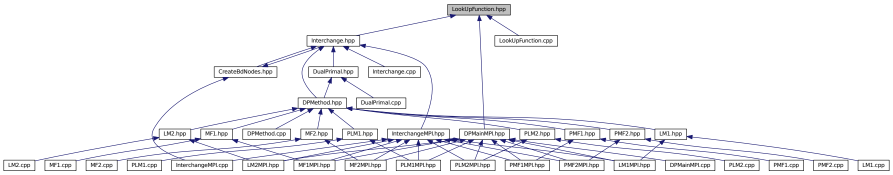
8.53 LookUpFunction.hpp File Reference

```
#include <string>
#include "FunctionV1.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for LookUpFunction.hpp:



This graph shows which files directly or indirectly include this file:



Classes

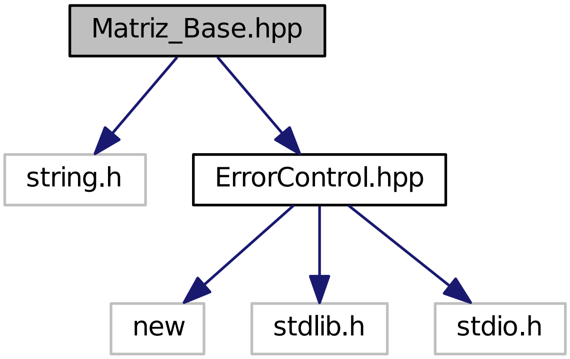
- class LookUpFunction

8.54 main.hpp File Reference

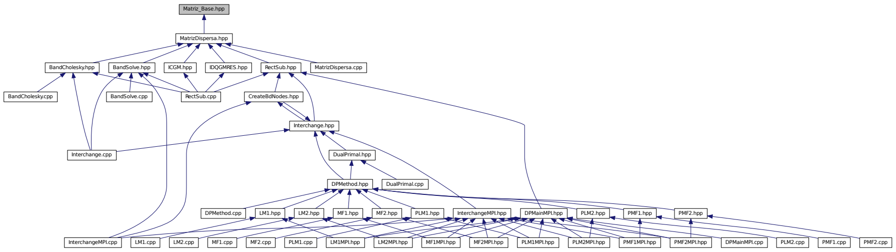
8.55 Matriz_Base.hpp File Reference

```
#include <string.h>
#include "ErrorControl.hpp"
```

Include dependency graph for Matriz_Base.hpp:



This graph shows which files directly or indirectly include this file:



Classes

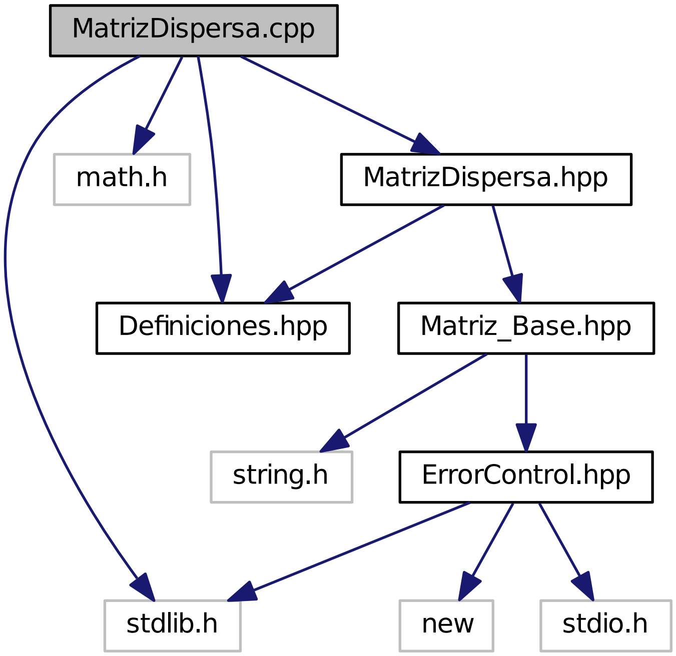
- class Matriz_Base

    Clase base para el trabajar con matrices.

8.56 MatrizDispersa.cpp File Reference

```
#include <stdlib.h>
#include <math.h>
#include "Definiciones.hpp"
#include "MatrizDispersa.hpp"
```

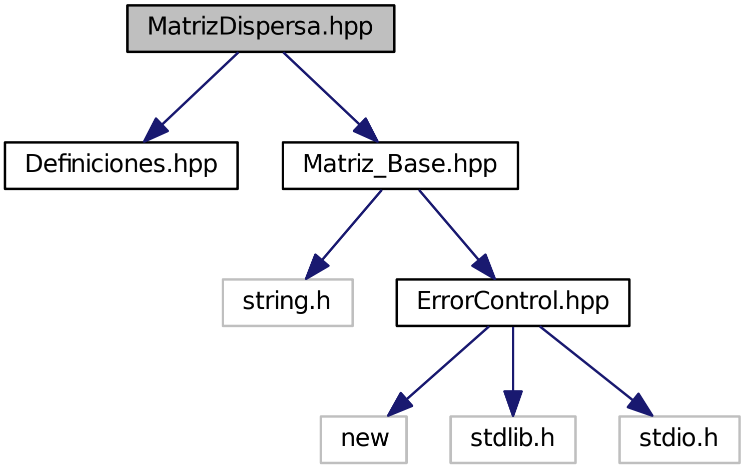
Include dependency graph for MatrizDispersa.cpp:



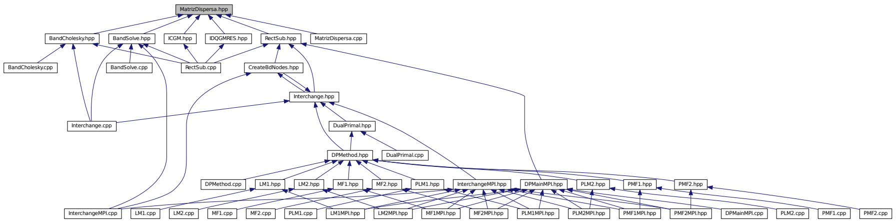
8.57 MatrizDispersa.hpp File Reference

```
#include "Definiciones.hpp"
#include "Matriz_Base.hpp"
```

Include dependency graph for MatrizDispersa.hpp:



This graph shows which files directly or indirectly include this file:



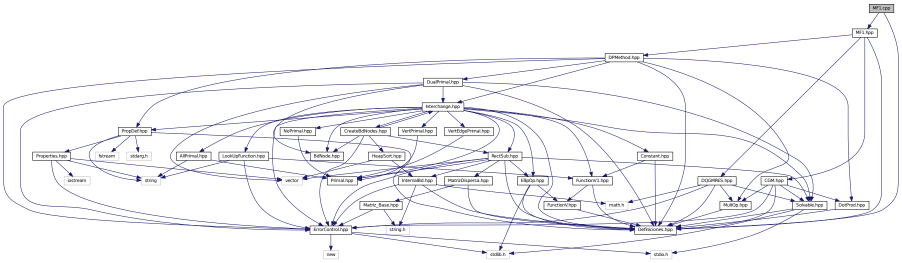
Classes

- class MatrizDispersa

8.58 MF1.cpp File Reference

```
#include "Definiciones.hpp"
#include "MF1.hpp"
```

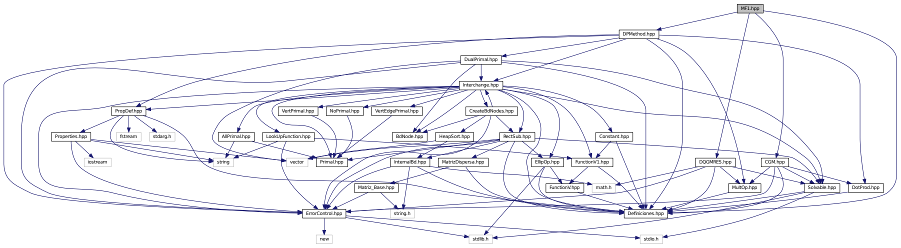
Include dependency graph for MF1.cpp:



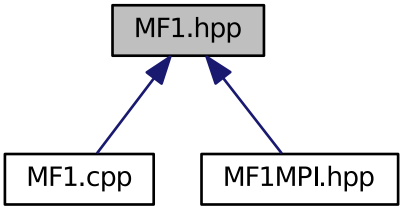
8.59 MF1.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for MF1.hpp:



This graph shows which files directly or indirectly include this file:



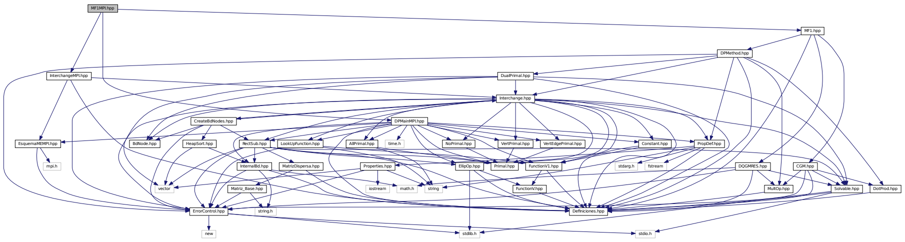
Classes

- class MF1

8.60 MF1MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "MF1.hpp"
```

Include dependency graph for MF1MPI.hpp:



Classes

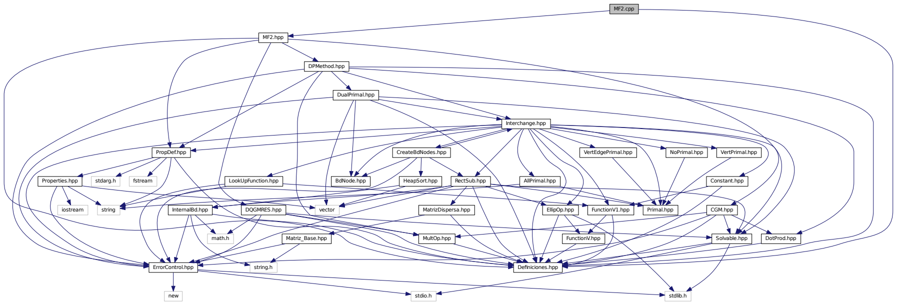
- class MF1MPI

  Clase para definir el metodo MF-1 de DVS-DDM.

8.61 MF2.cpp File Reference

```
#include "Definiciones.hpp"
#include "MF2.hpp"
```

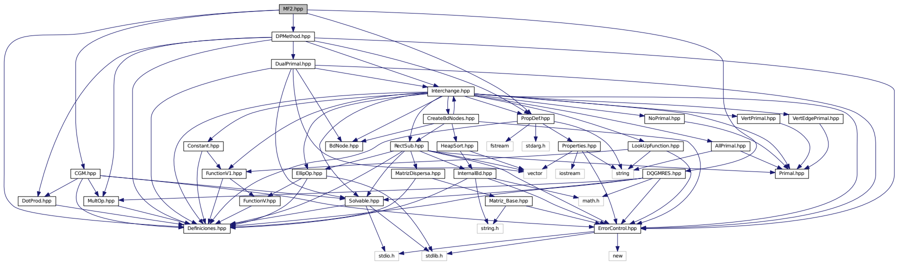
Include dependency graph for MF2.cpp:



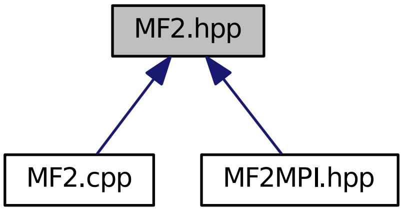
8.62 MF2.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "PropDef.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for MF2.hpp:



This graph shows which files directly or indirectly include this file:



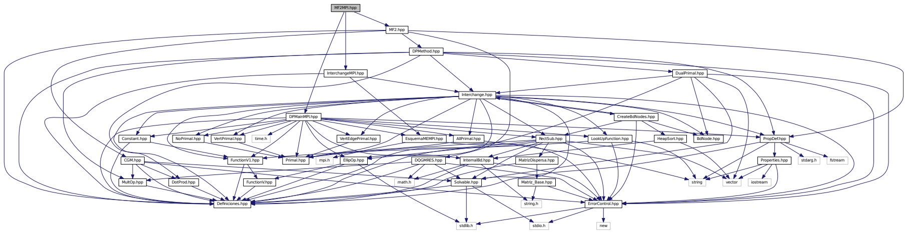
Classes

- class MF2

8.63 MF2MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "MF2.hpp"
```

Include dependency graph for MF2MPI.hpp:



Classes

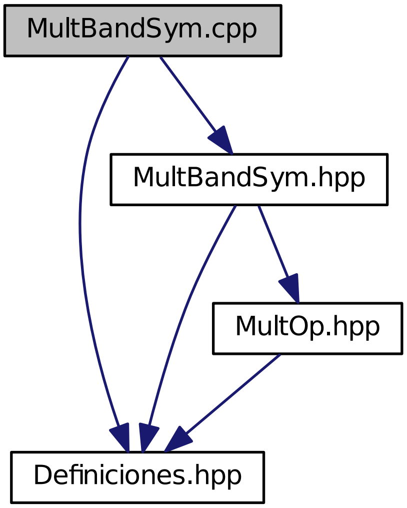
- class MF2MPI

    Clase para definir el metodo MF-2 de DVS-DDM.

8.64 MultBandSym.cpp File Reference

```
#include "Definiciones.hpp"
#include "MultBandSym.hpp"
```

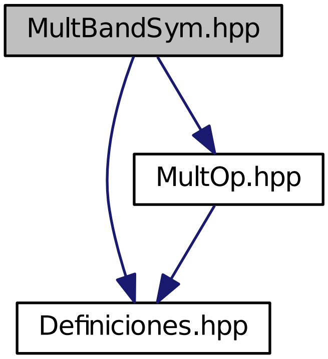
Include dependency graph for MultBandSym.cpp:



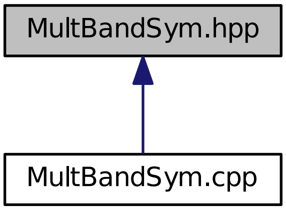
8.65 MultBandSym.hpp File Reference

```
#include "Definiciones.hpp"
#include "MultOp.hpp"
```

Include dependency graph for MultBandSym.hpp:



This graph shows which files directly or indirectly include this file:



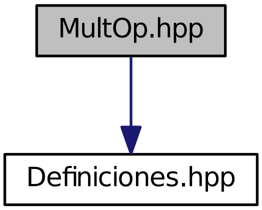
Classes

- class MultBandSym

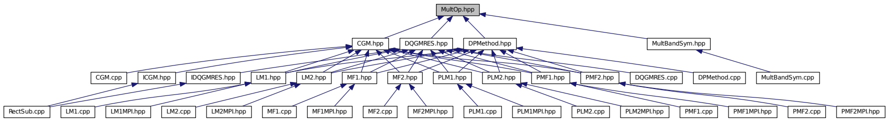
8.66 MultOp.hpp File Reference

```
#include "Definiciones.hpp"
```

Include dependency graph for MultOp.hpp:



This graph shows which files directly or indirectly include this file:



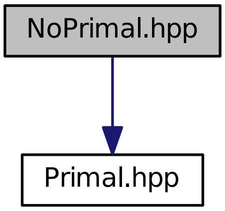
Classes

- class MultOp

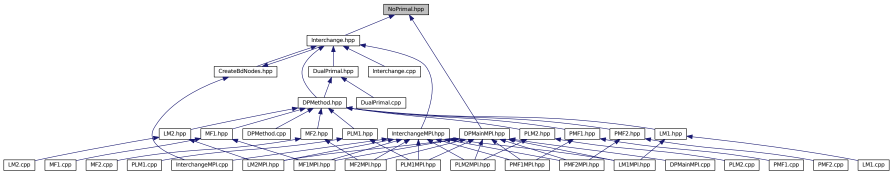
8.67 NoPrimal.hpp File Reference

```
#include "Primal.hpp"
```

Include dependency graph for NoPrimal.hpp:



This graph shows which files directly or indirectly include this file:



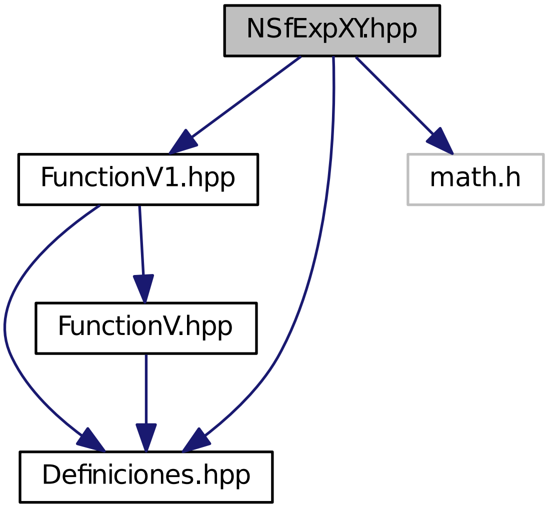
Classes

- class NoPrimal

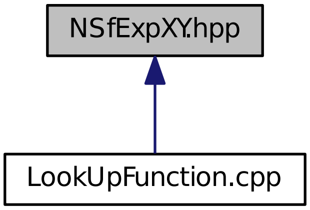
8.68 NSfExpXY.hpp File Reference

```
#include "FunctionV1.hpp"
#include "Definiciones.hpp"
#include <math.h>
```

Include dependency graph for NSfExpXY.hpp:



This graph shows which files directly or indirectly include this file:



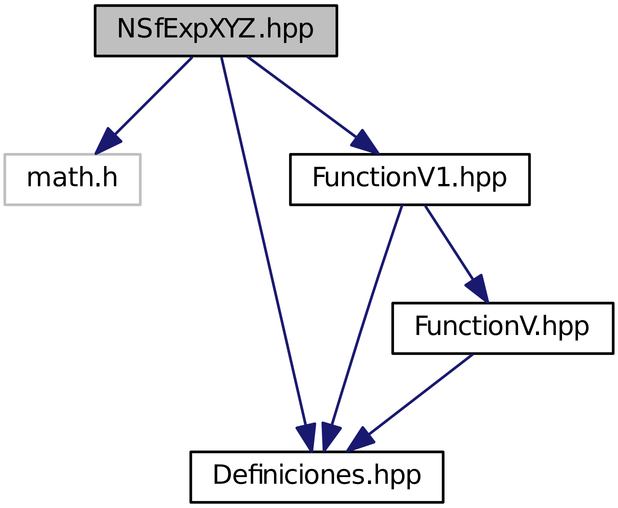
Classes

- class NSfExpXY

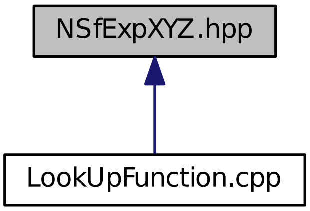
8.69 NSfExpXYZ.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for NSfExpXYZ.hpp:



This graph shows which files directly or indirectly include this file:



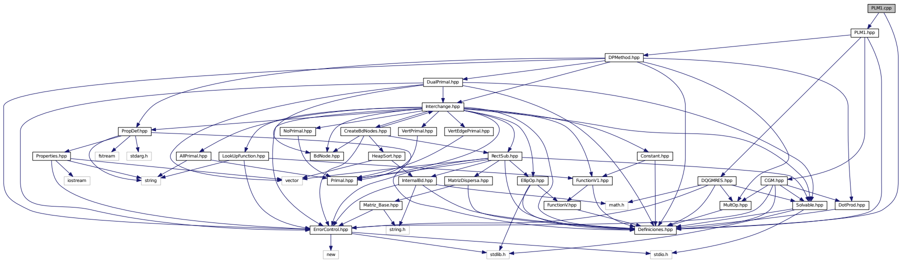
Classes

- class NSfExpXYZ

8.70 PLM1.cpp File Reference

```
#include "Definiciones.hpp"
#include "PLM1.hpp"
```

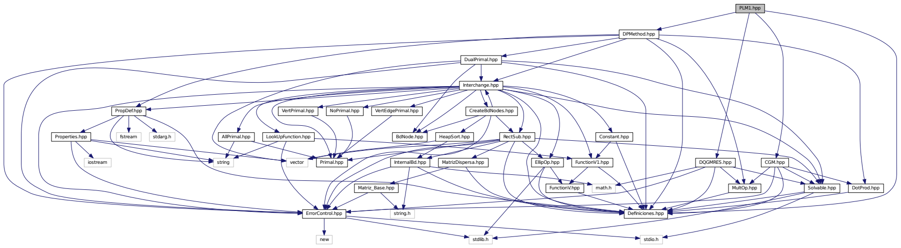
Include dependency graph for PLM1.cpp:



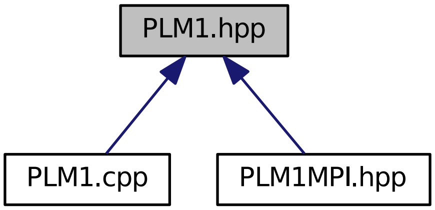
8.71 PLM1.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for PLM1.hpp:



This graph shows which files directly or indirectly include this file:



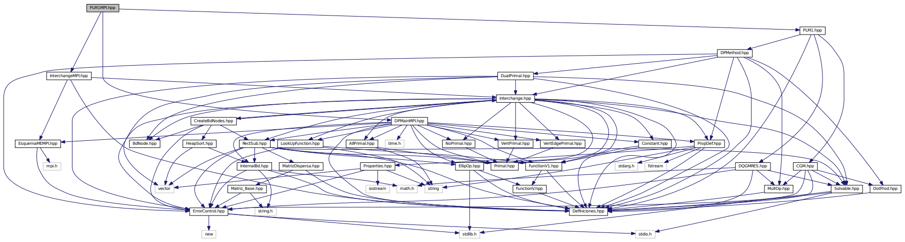
Classes

- class PLM1

8.72 PLM1MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PLM1.hpp"
```

Include dependency graph for PLM1MPI.hpp:



Classes

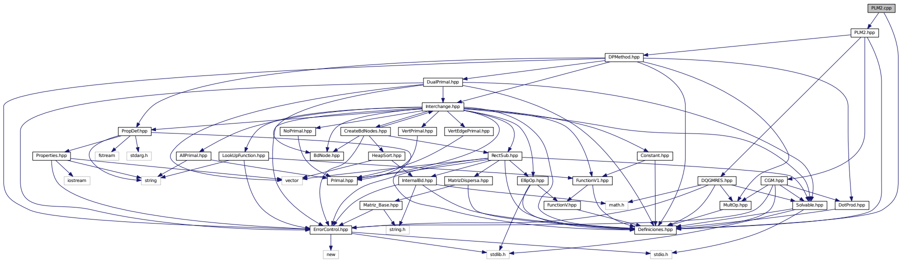
- class PLM1MPI

  Clase para definir el metodo PLM-1 de DVS-DDM.

8.73 PLM2.cpp File Reference

```
#include "Definiciones.hpp"
#include "PLM2.hpp"
```

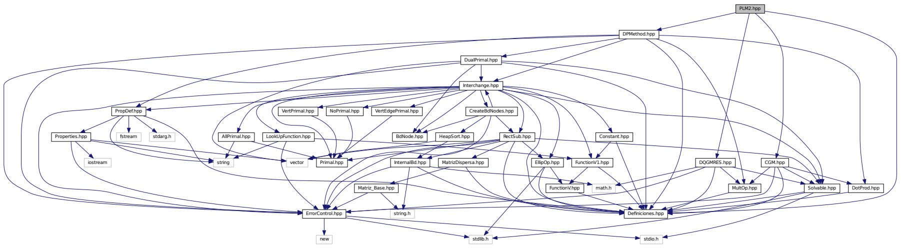
Include dependency graph for PLM2.cpp:



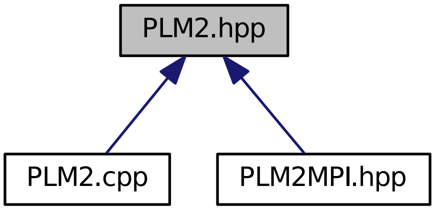
8.74 PLM2.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for PLM2.hpp:



This graph shows which files directly or indirectly include this file:



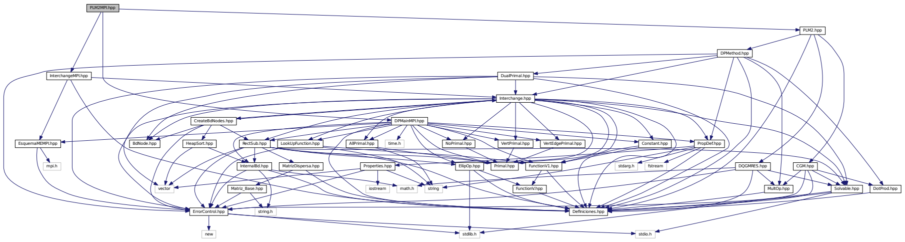
Classes

- class PLM2

8.75 PLM2MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PLM2.hpp"
```

Include dependency graph for PLM2MPI.hpp:



Classes

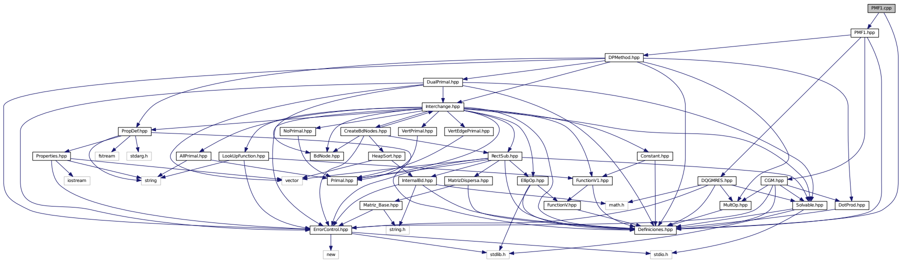
- class PLM2MPI

  Clase para definir el metodo MF-1 de DVS-DDM.

8.76 PMF1.cpp File Reference

```
#include "Definiciones.hpp"
#include "PMF1.hpp"
```

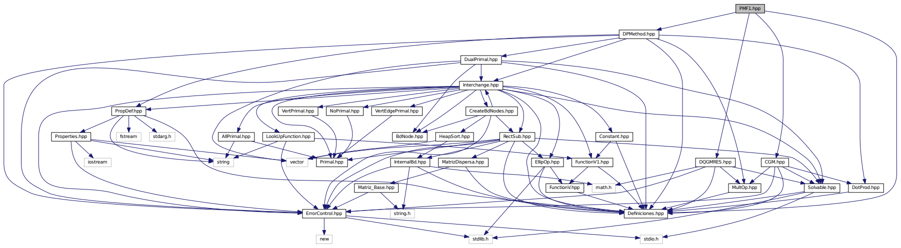
Include dependency graph for PMF1.cpp:



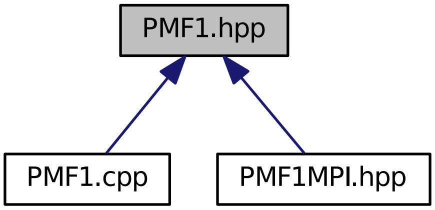
8.77 PMF1.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for PMF1.hpp:



This graph shows which files directly or indirectly include this file:



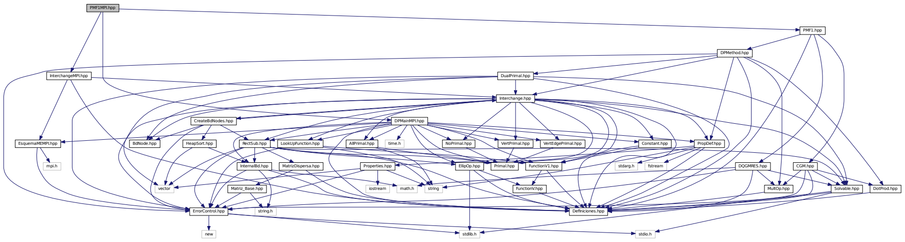
Classes

- class PMF1

8.78 PMF1MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PMF1.hpp"
```

Include dependency graph for PMF1MPI.hpp:



Classes

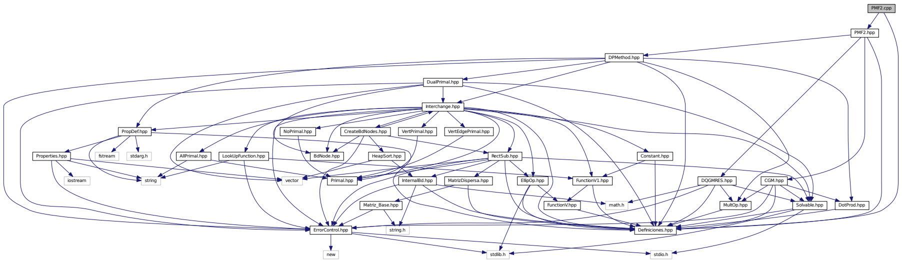
- class PMF1MPI

  Clase para definir el metodo PMF-1 de DVS-DDM.

8.79 PMF2.cpp File Reference

```
#include "Definiciones.hpp"
#include "PMF2.hpp"
```

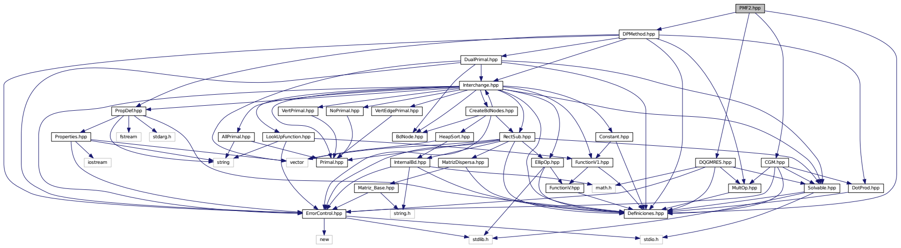
Include dependency graph for PMF2.cpp:



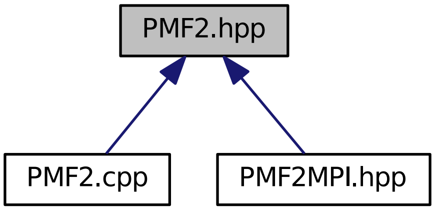
8.80 PMF2.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for PMF2.hpp:



This graph shows which files directly or indirectly include this file:



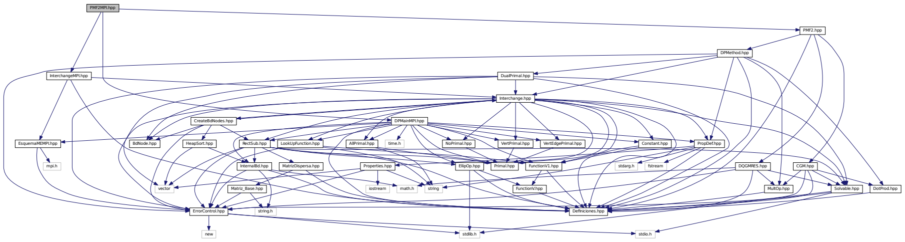
Classes

- class PMF2

8.81 PMF2MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PMF2.hpp"
```

Include dependency graph for PMF2MPI.hpp:



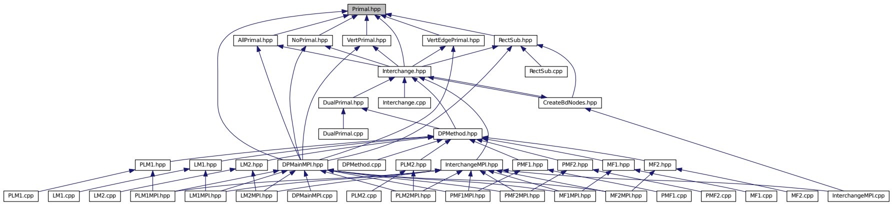
Classes

- class PMF2MPI

  Clase para definir el metodo PMF-2 de DVS-DDM.

8.82 Primal.hpp File Reference

This graph shows which files directly or indirectly include this file:



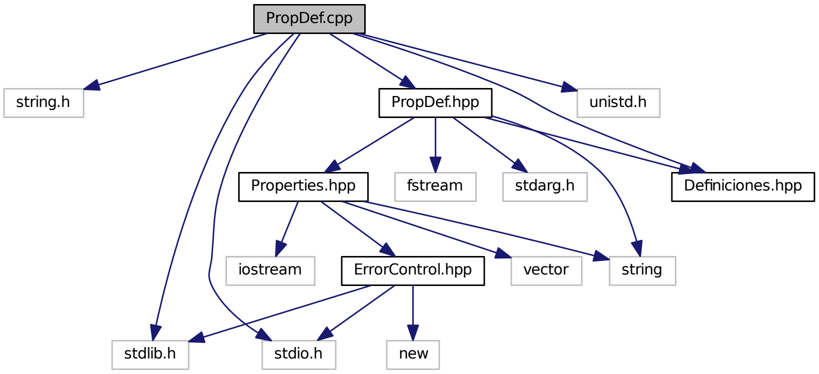
Classes

- class Primal

8.83 PropDef.cpp File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include "Definiciones.hpp"
#include "PropDef.hpp"
```

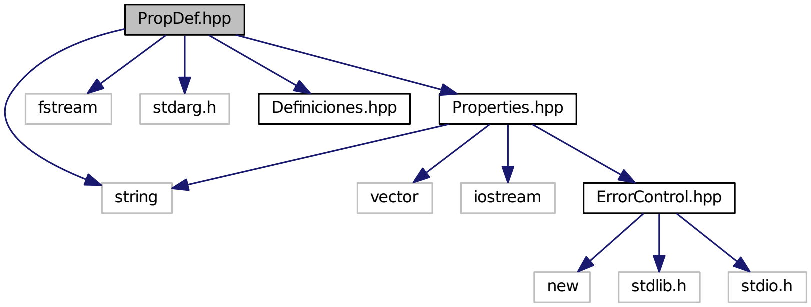
Include dependency graph for PropDef.cpp:



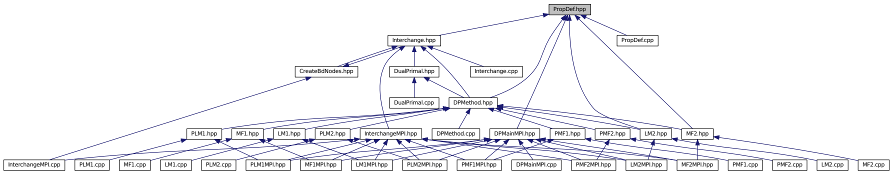
8.84 PropDef.hpp File Reference

```
#include <string>
#include <fstream>
#include <stdarg.h>
#include "Definiciones.hpp"
#include "Properties.hpp"
```

Include dependency graph for PropDef.hpp:



This graph shows which files directly or indirectly include this file:



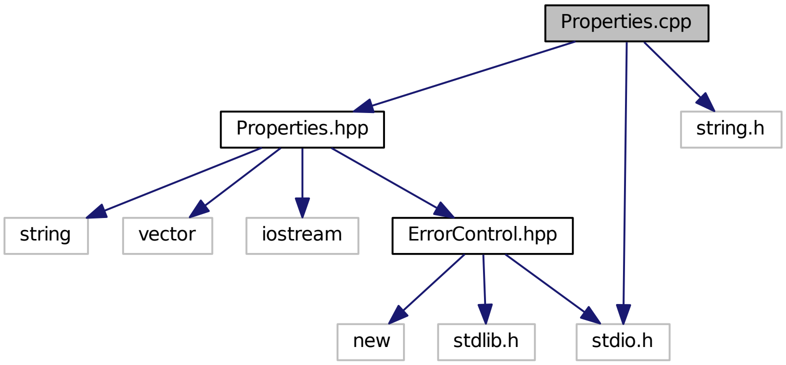
Classes

- class PropDef

8.85 Properties.cpp File Reference

```
#include "Properties.hpp"
#include <string.h>
#include <stdio.h>
```

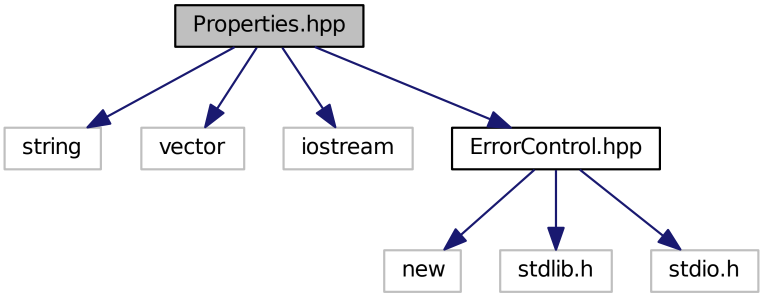
Include dependency graph for Properties.cpp:



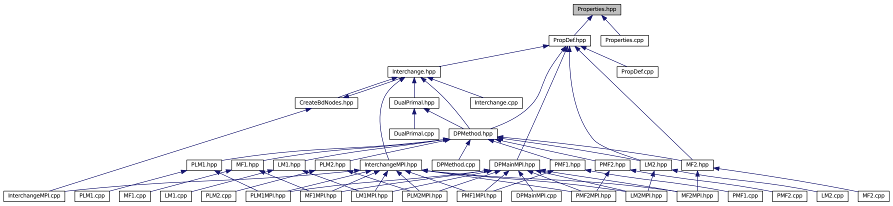
8.86 Properties.hpp File Reference

```
#include <string>
#include <vector>
#include <iostream>
#include "ErrorControl.hpp"
```

Include dependency graph for Properties.hpp:



This graph shows which files directly or indirectly include this file:



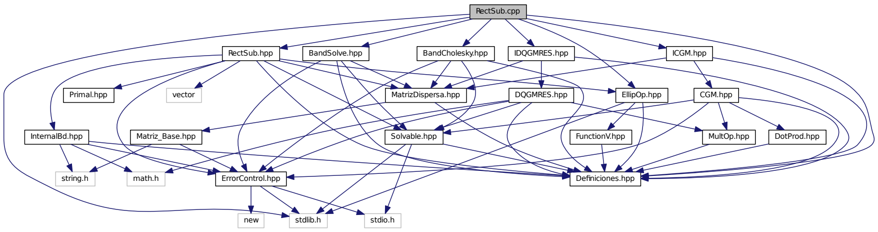
Classes

- class Properties

8.87 RectSub.cpp File Reference

```
#include <stdlib.h>
#include "Definiciones.hpp"
#include "RectSub.hpp"
#include "EllipOp.hpp"
#include "ICGM.hpp"
#include "IDQGMRES.hpp"
#include "BandSolve.hpp"
#include "BandCholesky.hpp"
```

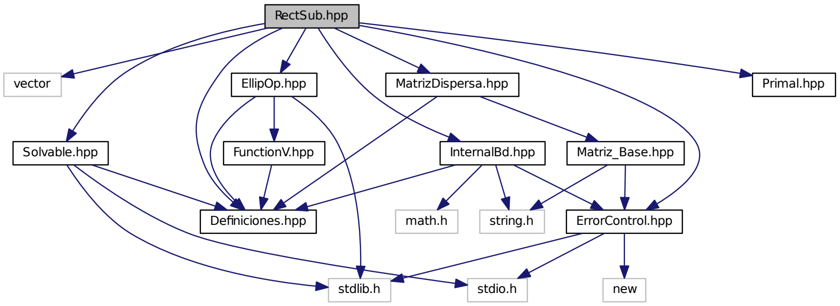
Include dependency graph for RectSub.cpp:



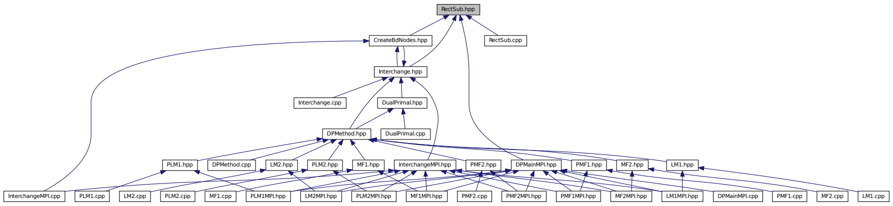
8.88 RectSub.hpp File Reference

```
#include <vector>
#include "Definiciones.hpp"
#include "InternalBd.hpp"
#include "EllipOp.hpp"
#include "Solvable.hpp"
#include "Primal.hpp"
#include "MatrizDispersa.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for RectSub.hpp:



This graph shows which files directly or indirectly include this file:



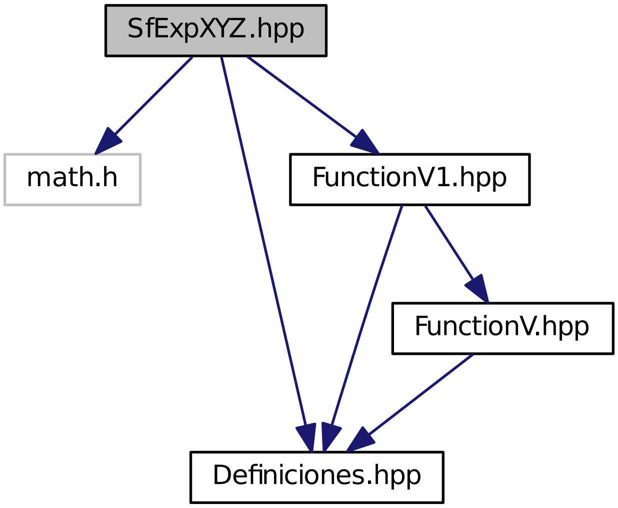
Classes

- class RectSub

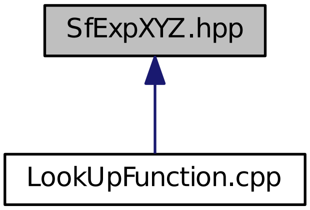
8.89 SfExpXYZ.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for SfExpXYZ.hpp:



This graph shows which files directly or indirectly include this file:



Classes

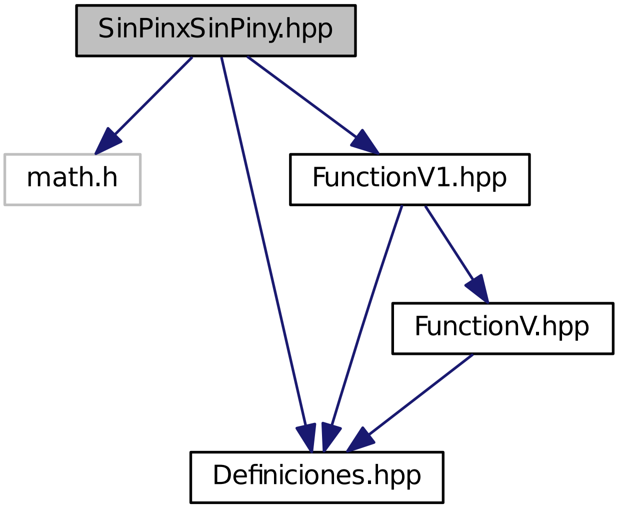
- class SfExpXYZ

8.90 SinPinxSinPiny.hpp File Reference

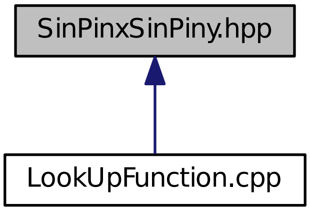
```
#include <math.h>
```

```
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for SinPinxSinPiny.hpp:



This graph shows which files directly or indirectly include this file:



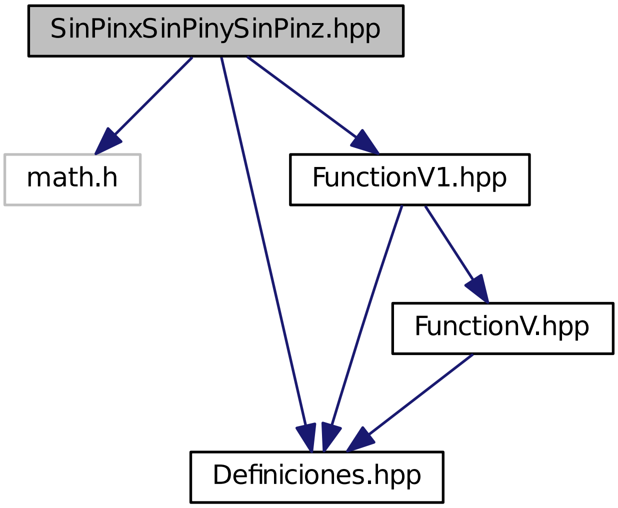
Classes

- class SinPinxSinPiny

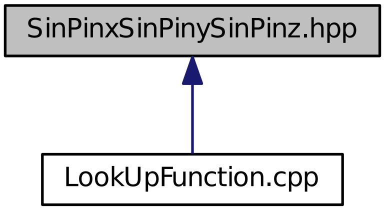
8.91 SinPinxSinPinySinPinz.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for SinPinxSinPinySinPinz.hpp:



This graph shows which files directly or indirectly include this file:



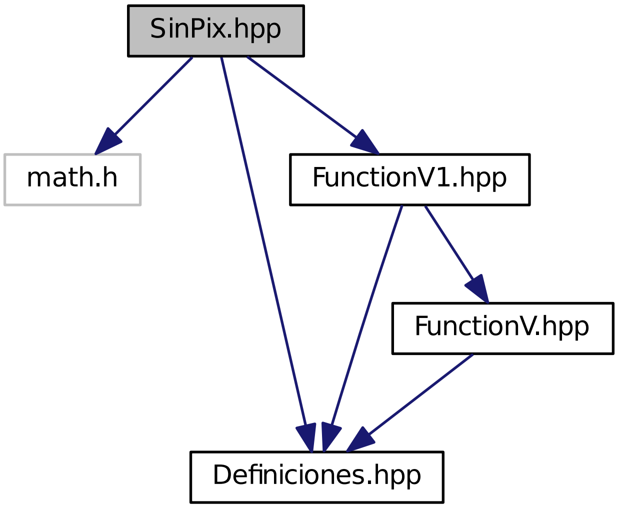
Classes

- class SinPinxSinPinySinPinz

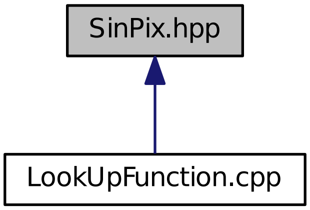
8.92 SinPix.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for SinPix.hpp:



This graph shows which files directly or indirectly include this file:



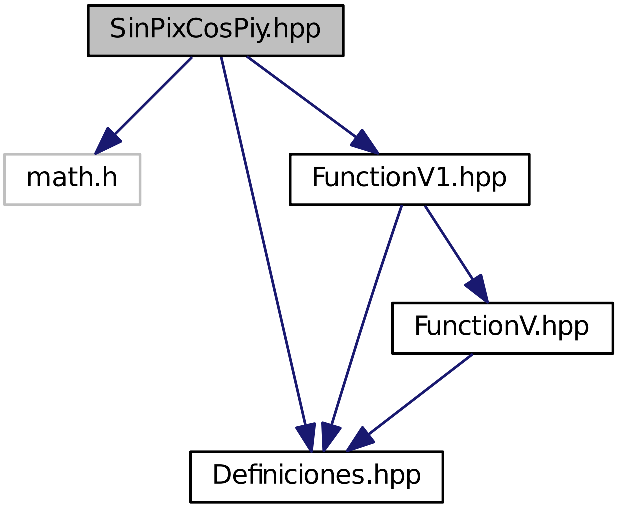
Classes

- class SinPix

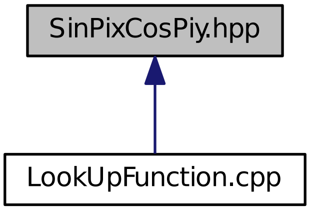
8.93 SinPixCosPiy.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for SinPixCosPiy.hpp:



This graph shows which files directly or indirectly include this file:



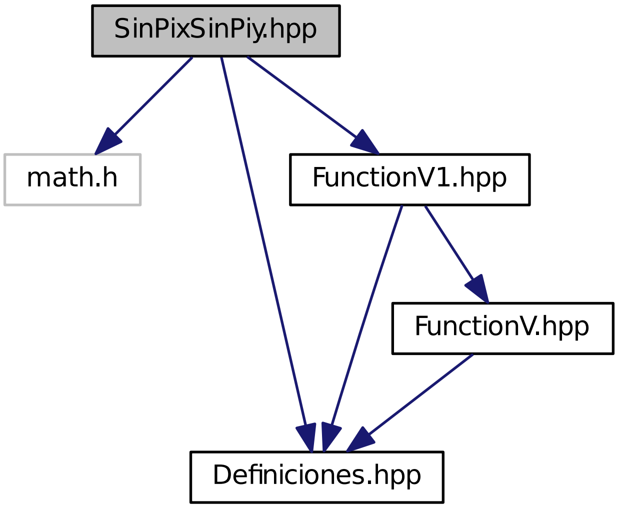
Classes

- class SinPixCosPiy

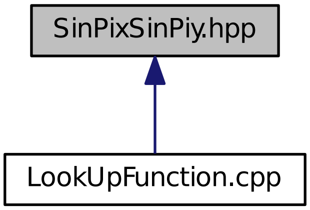
8.94 SinPixSinPiy.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for SinPixSinPiy.hpp:



This graph shows which files directly or indirectly include this file:



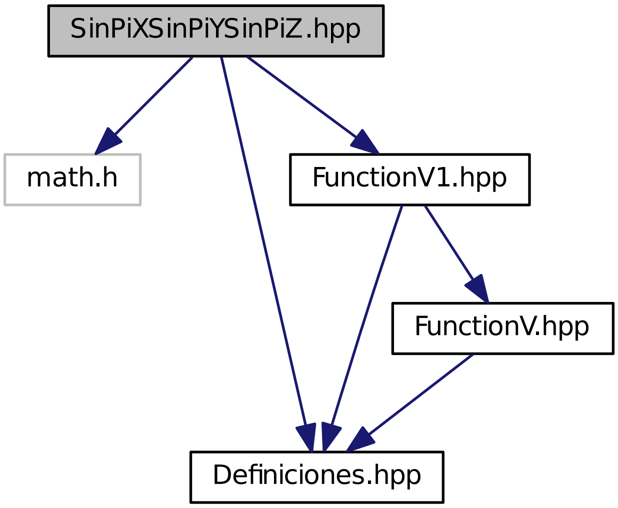
Classes

- class SinPixSinPiy

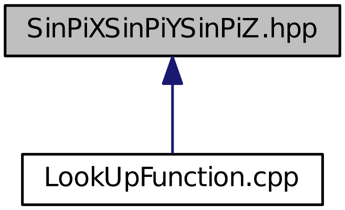
8.95 SinPiXSinPiYSinPiZ.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for SinPiXSinPiYSinPiZ.hpp:



This graph shows which files directly or indirectly include this file:



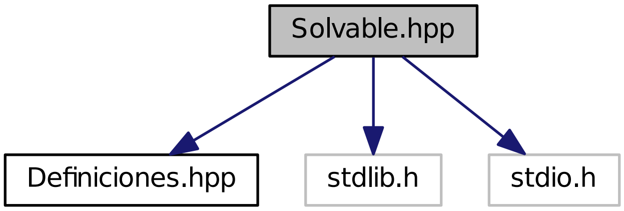
Classes

- class SinPiXSinPiYSinPiZ

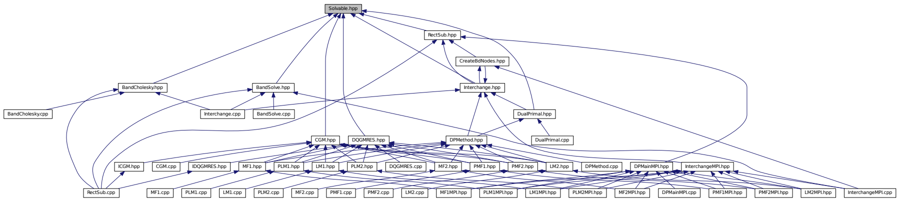
8.96 Solvable.hpp File Reference

```
#include "Definiciones.hpp"
#include <stdlib.h>
#include <stdio.h>
```

Include dependency graph for Solvable.hpp:



This graph shows which files directly or indirectly include this file:



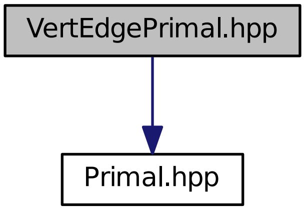
Classes

- class Solvable

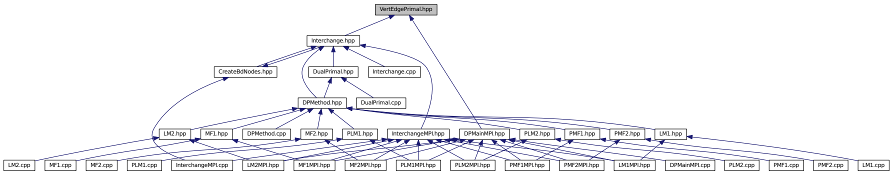
8.97 VertEdgePrimal.hpp File Reference

```
#include "Primal.hpp"
```

Include dependency graph for VertEdgePrimal.hpp:



This graph shows which files directly or indirectly include this file:



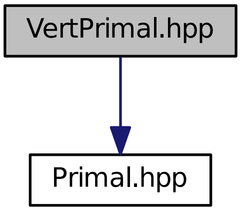
Classes

- class VertEdgePrimal

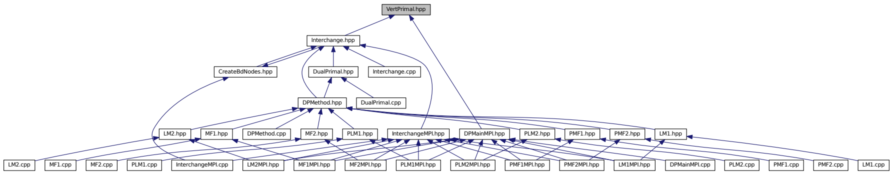
8.98 VertPrimal.hpp File Reference

```
#include "Primal.hpp"
```

Include dependency graph for VertPrimal.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class VertPrimal

Chapter 9

Example Documentation

9.1 EjemploMatrizDispersa.cpp

Esta clase implementa los componentes para el trabajar con matrices dispersas de punto flotante

9.2 ExampleErrorControl.cpp

Error Control.

Index