

DDM-DVS

Generated by Doxygen 1.8.1.2

Thu May 9 2013 08:35:41

Contents

1	Grupo de Modelación Matemática y Computacional, UNAM	1
1.1	Introducción	1
2	Todo List	3
3	Bug List	5
4	Class Index	7
4.1	Class Hierarchy	7
5	Class Index	9
5.1	Class List	9
6	File Index	11
6.1	File List	11
7	Class Documentation	15
7.1	AllPrimal Class Reference	15
7.1.1	Constructor & Destructor Documentation	17
7.1.1.1	AllPrimal	17
7.1.2	Member Function Documentation	17
7.1.2.1	isPrimal	17
7.1.3	Member Data Documentation	17
7.1.3.1	name	17
7.2	BandCholesky Class Reference	17
7.2.1	Constructor & Destructor Documentation	20
7.2.1.1	BandCholesky	20
7.2.1.2	~BandCholesky	20
7.2.2	Member Function Documentation	20
7.2.2.1	clean	20
7.2.2.2	convertBand	20

7.2.2.3	convertBand	20
7.2.2.4	factorLU	20
7.2.2.5	getIter	20
7.2.2.6	print	20
7.2.2.7	solve	20
7.2.3	Member Data Documentation	21
7.2.3.1	AK	21
7.2.3.2	bw	21
7.2.3.3	ce	21
7.2.3.4	n	21
7.3	BandSolve Class Reference	21
7.3.1	Constructor & Destructor Documentation	24
7.3.1.1	BandSolve	24
7.3.1.2	BandSolve	24
7.3.1.3	BandSolve	24
7.3.1.4	~BandSolve	24
7.3.2	Member Function Documentation	24
7.3.2.1	clean	24
7.3.2.2	convertBand	24
7.3.2.3	convertBand	24
7.3.2.4	factorLU	24
7.3.2.5	getIter	24
7.3.2.6	print	25
7.3.2.7	solve	25
7.3.3	Member Data Documentation	25
7.3.3.1	AK	25
7.3.3.2	bw	25
7.3.3.3	ce	25
7.3.3.4	n	25
7.4	BdNode Class Reference	25
7.4.1	Constructor & Destructor Documentation	26
7.4.1.1	BdNode	26
7.4.2	Member Data Documentation	26
7.4.2.1	index	26
7.4.2.2	mult	26
7.4.2.3	node	27
7.4.2.4	subd	27

7.5	CGM Class Reference	27
7.5.1	Constructor & Destructor Documentation	30
7.5.1.1	CGM	30
7.5.1.2	CGM	30
7.5.1.3	~CGM	30
7.5.2	Member Function Documentation	30
7.5.2.1	clean	30
7.5.2.2	getIter	30
7.5.2.3	inicializa	31
7.5.2.4	norm	31
7.5.2.5	setEpsilon	31
7.5.2.6	setMaxIter	31
7.5.2.7	solve	31
7.5.3	Member Data Documentation	31
7.5.3.1	A	31
7.5.3.2	alpha	31
7.5.3.3	beta	31
7.5.3.4	ce	31
7.5.3.5	dotP	31
7.5.3.6	eps	31
7.5.3.7	gamma	31
7.5.3.8	lambda	31
7.5.3.9	mu	31
7.5.3.10	n	31
7.5.3.11	nIter	31
7.5.3.12	nMaxIter	31
7.5.3.13	p	31
7.5.3.14	r	31
7.5.3.15	v	31
7.6	Constant Class Reference	32
7.6.1	Constructor & Destructor Documentation	34
7.6.1.1	Constant	34
7.6.2	Member Function Documentation	34
7.6.2.1	eval	34
7.6.2.2	getVar	34
7.6.2.3	setVar	34
7.6.3	Member Data Documentation	34

7.6.3.1	a	34
7.7	CreateBdNodes Class Reference	34
7.7.1	Constructor & Destructor Documentation	36
7.7.1.1	CreateBdNodes	36
7.7.1.2	~CreateBdNodes	36
7.7.2	Member Data Documentation	36
7.7.2.1	bdAll	36
7.7.2.2	bdDuals	36
7.7.2.3	bdPrimals	36
7.7.2.4	ce	36
7.7.2.5	dualMult	36
7.7.2.6	hbd	36
7.7.2.7	ibd	36
7.7.2.8	ibdAll	36
7.7.2.9	maxBd	36
7.7.2.10	nD	36
7.7.2.11	nDual	36
7.7.2.12	nP	36
7.7.2.13	nPrimal	37
7.8	Disc Class Reference	37
7.8.1	Constructor & Destructor Documentation	39
7.8.1.1	Disc	39
7.8.2	Member Function Documentation	39
7.8.2.1	eval	39
7.8.2.2	getVar	39
7.8.2.3	setVar	39
7.8.3	Member Data Documentation	39
7.8.3.1	var	39
7.9	Disc13 Class Reference	39
7.9.1	Constructor & Destructor Documentation	42
7.9.1.1	Disc13	42
7.9.2	Member Function Documentation	42
7.9.2.1	eval	42
7.9.2.2	getVar	42
7.9.2.3	setVar	42
7.9.3	Member Data Documentation	42
7.9.3.1	var	42

7.10	Disc14 Class Reference	42
7.10.1	Constructor & Destructor Documentation	45
7.10.1.1	Disc14	45
7.10.2	Member Function Documentation	45
7.10.2.1	eval	45
7.10.2.2	getVar	45
7.10.2.3	setVar	45
7.10.3	Member Data Documentation	45
7.10.3.1	var	45
7.11	Disc15 Class Reference	45
7.11.1	Constructor & Destructor Documentation	48
7.11.1.1	Disc15	48
7.11.2	Member Function Documentation	48
7.11.2.1	eval	48
7.11.2.2	getVar	48
7.11.2.3	setVar	48
7.11.3	Member Data Documentation	48
7.11.3.1	var	48
7.12	DotProd Class Reference	48
7.12.1	Member Function Documentation	49
7.12.1.1	dot	50
7.13	DPMainMPI Class Reference	50
7.13.1	Detailed Description	52
7.13.2	Constructor & Destructor Documentation	53
7.13.2.1	DPMainMPI	53
7.13.2.2	~DPMainMPI	53
7.13.3	Member Function Documentation	53
7.13.3.1	deleteInternalBd	53
7.13.3.2	Esclavo	53
7.13.4	Member Data Documentation	53
7.13.4.1	c	53
7.13.4.2	domain	53
7.13.4.3	f	53
7.13.4.4	fc	53
7.13.4.5	g	53
7.13.4.6	gc	53
7.13.4.7	hbd	53

7.13.4.8	indl	53
7.13.4.9	mesh	54
7.13.4.10	method	54
7.13.4.11	msa	54
7.13.4.12	mss	54
7.13.4.13	nDim	54
7.13.4.14	nOmega	54
7.13.4.15	nta	54
7.13.4.16	omegas	54
7.13.4.17	one	54
7.13.4.18	op	54
7.13.4.19	prim	54
7.13.4.20	primal	54
7.13.4.21	sf	54
7.13.4.22	sg	54
7.13.4.23	swprint	54
7.13.4.24	t1	54
7.13.4.25	t2	54
7.13.4.26	xnp	54
7.13.4.27	zero	55
7.14	DPMethod Class Reference	55
7.14.1	Constructor & Destructor Documentation	57
7.14.1.1	DPMethod	57
7.14.1.2	~DPMethod	57
7.14.2	Member Function Documentation	57
7.14.2.1	analyticSolution	58
7.14.2.2	clean	58
7.14.2.3	conditionalNumber	58
7.14.2.4	genInverse	58
7.14.2.5	getSize	58
7.14.2.6	iniInterchage	58
7.14.2.7	initialize	58
7.14.2.8	prCoord	58
7.14.2.9	print	58
7.14.2.10	printTime	58
7.14.2.11	rhs	58
7.14.2.12	solve	58

7.14.3	Member Data Documentation	58
7.14.3.1	ce	58
7.14.3.2	dualp	59
7.14.3.3	epsilon	59
7.14.3.4	inter	59
7.14.3.5	nDim	59
7.14.3.6	nDual	59
7.14.3.7	nOmega	59
7.14.3.8	printv	59
7.14.3.9	props	59
7.14.3.10	rhss	59
7.14.3.11	scr	59
7.14.3.12	solver	59
7.14.3.13	time0	59
7.14.3.14	time1	59
7.14.3.15	time2	59
7.14.3.16	time3	59
7.14.3.17	u	59
7.15	DQGMRES Class Reference	59
7.15.1	Constructor & Destructor Documentation	62
7.15.1.1	DQGMRES	62
7.15.1.2	DQGMRES	62
7.15.1.3	~DQGMRES	62
7.15.2	Member Function Documentation	62
7.15.2.1	applyOmega	62
7.15.2.2	clean	62
7.15.2.3	getIter	62
7.15.2.4	inicializa	63
7.15.2.5	setEpsilon	63
7.15.2.6	setMaxIter	63
7.15.2.7	solve	63
7.15.3	Member Data Documentation	63
7.15.3.1	ce	63
7.15.3.2	cs	63
7.15.3.3	eps	63
7.15.3.4	gm	63
7.15.3.5	gm1	63

7.15.3.6	h	63
7.15.3.7	k	63
7.15.3.8	k1	63
7.15.3.9	maxIter	63
7.15.3.10	mult	63
7.15.3.11	n	63
7.15.3.12	nIter	63
7.15.3.13	nMaxIter	63
7.15.3.14	p	63
7.15.3.15	q	63
7.15.3.16	v	63
7.16	DualPrimal Class Reference	64
7.16.1	Constructor & Destructor Documentation	65
7.16.1.1	DualPrimal	65
7.16.1.2	~DualPrimal	65
7.16.2	Member Function Documentation	65
7.16.2.1	a	65
7.16.2.2	calcValues	65
7.16.2.3	fromSubdomains	65
7.16.2.4	genMats	65
7.16.2.5	getNDual	65
7.16.2.6	j	65
7.16.2.7	multS	65
7.16.2.8	solveAPP	66
7.16.2.9	solveS	66
7.16.2.10	toSubdomains	66
7.16.3	Member Data Documentation	66
7.16.3.1	ce	66
7.16.3.2	inter	66
7.16.3.3	nD	66
7.16.3.4	nDual	66
7.16.3.5	nP	66
7.16.3.6	nPrimal	66
7.16.3.7	SP	66
7.16.3.8	XP	66
7.16.3.9	YP	66
7.17	EllipOp Class Reference	66

7.17.1	Constructor & Destructor Documentation	68
7.17.1.1	EllipOp	68
7.17.1.2	EllipOp	68
7.17.1.3	~EllipOp	68
7.17.2	Member Function Documentation	68
7.17.2.1	getA	68
7.17.2.2	getB	68
7.17.2.3	getC	68
7.17.2.4	getF	68
7.17.2.5	getG	68
7.17.2.6	isSymmetric	68
7.17.2.7	setF	68
7.17.2.8	setG	68
7.17.3	Member Data Documentation	68
7.17.3.1	a	68
7.17.3.2	b	69
7.17.3.3	c	69
7.17.3.4	f	69
7.17.3.5	g	69
7.17.3.6	nDim	69
7.17.3.7	sol	69
7.18	ErrorControl Class Reference	69
7.18.1	Detailed Description	71
7.18.2	Constructor & Destructor Documentation	71
7.18.2.1	ErrorControl	71
7.18.2.2	ErrorControl	71
7.18.2.3	ErrorControl	71
7.18.3	Member Function Documentation	72
7.18.3.1	fatalError	72
7.18.3.2	fatalError	72
7.18.3.3	memoryError	72
7.18.3.4	memoryError	72
7.18.3.5	memoryError	72
7.18.3.6	nameClass	72
7.18.3.7	nameClassFunct	73
7.18.3.8	nameFunct	73
7.18.4	Member Data Documentation	73

7.18.4.1	nmClass	73
7.18.4.2	nmFunction	73
7.19	EsquemaMEMPI Class Reference	73
7.19.1	Detailed Description	76
7.19.2	Constructor & Destructor Documentation	77
7.19.2.1	EsquemaMEMPI	77
7.19.2.2	~EsquemaMEMPI	77
7.19.3	Member Function Documentation	77
7.19.3.1	generaRepartoCarga	77
7.19.3.2	numeroProcesadoresUsar	77
7.19.3.3	reparteCargaTrabajo	77
7.19.4	Member Data Documentation	77
7.19.4.1	ce	77
7.19.4.2	id	78
7.19.4.3	np	78
7.19.4.4	npu	78
7.19.4.5	ta	78
7.20	ExpVXY Class Reference	78
7.20.1	Constructor & Destructor Documentation	81
7.20.1.1	ExpVXY	81
7.20.2	Member Function Documentation	81
7.20.2.1	eval	81
7.20.2.2	getVar	81
7.20.2.3	setVar	81
7.20.3	Member Data Documentation	81
7.20.3.1	var	81
7.21	ExpVXYZ Class Reference	81
7.21.1	Constructor & Destructor Documentation	84
7.21.1.1	ExpVXYZ	84
7.21.2	Member Function Documentation	84
7.21.2.1	eval	84
7.21.2.2	getVar	84
7.21.2.3	setVar	84
7.21.3	Member Data Documentation	84
7.21.3.1	var	84
7.22	ExpX Class Reference	84
7.22.1	Constructor & Destructor Documentation	87

7.22.1.1	ExpX	87
7.22.2	Member Function Documentation	87
7.22.2.1	eval	87
7.22.2.2	getVar	87
7.22.2.3	setVar	87
7.22.3	Member Data Documentation	87
7.22.3.1	var	87
7.23	ExpXY Class Reference	87
7.23.1	Constructor & Destructor Documentation	90
7.23.1.1	ExpXY	90
7.23.2	Member Function Documentation	90
7.23.2.1	eval	90
7.23.2.2	getVar	90
7.23.2.3	setVar	90
7.23.3	Member Data Documentation	90
7.23.3.1	var	90
7.24	ExpXYZ Class Reference	90
7.24.1	Constructor & Destructor Documentation	93
7.24.1.1	ExpXYZ	93
7.24.2	Member Function Documentation	93
7.24.2.1	eval	93
7.24.2.2	getVar	93
7.24.2.3	setVar	93
7.24.3	Member Data Documentation	93
7.24.3.1	var	93
7.25	fExpXY Class Reference	93
7.25.1	Constructor & Destructor Documentation	96
7.25.1.1	fExpXY	96
7.25.2	Member Function Documentation	96
7.25.2.1	eval	96
7.25.2.2	getVar	96
7.25.2.3	setVar	96
7.25.3	Member Data Documentation	96
7.25.3.1	var	96
7.26	FunctionV Class Reference	96
7.26.1	Constructor & Destructor Documentation	97
7.26.1.1	FunctionV	97

7.26.1.2	~FunctionV	97
7.26.2	Member Function Documentation	97
7.26.2.1	dimension	97
7.26.2.2	eval	98
7.26.3	Member Data Documentation	98
7.26.3.1	dim	98
7.27	FunctionV1 Class Reference	98
7.27.1	Member Function Documentation	99
7.27.1.1	getVar	99
7.27.1.2	setVar	100
7.28	HeapSort Class Reference	100
7.28.1	Constructor & Destructor Documentation	101
7.28.1.1	HeapSort	101
7.28.1.2	~HeapSort	101
7.28.2	Member Function Documentation	101
7.28.2.1	genHeap	101
7.28.2.2	rr	101
7.28.2.3	siftup	101
7.28.2.4	sort	101
7.28.2.5	swap	101
7.28.3	Member Data Documentation	101
7.28.3.1	n	101
7.28.3.2	r	101
7.29	ICGM Class Reference	101
7.29.1	Detailed Description	104
7.29.2	Constructor & Destructor Documentation	104
7.29.2.1	ICGM	104
7.29.2.2	~ICGM	105
7.29.3	Member Function Documentation	105
7.29.3.1	clean	105
7.29.3.2	dot	105
7.29.3.3	getSize	105
7.29.3.4	multOp	105
7.29.4	Member Data Documentation	105
7.29.4.1	i	105
7.29.4.2	M	105
7.29.4.3	val	105

7.30	IDQGMRES Class Reference	105
7.30.1	Detailed Description	108
7.30.2	Constructor & Destructor Documentation	108
7.30.2.1	IDQGMRES	108
7.30.2.2	~IDQGMRES	108
7.30.3	Member Function Documentation	108
7.30.3.1	clean	108
7.30.3.2	getSize	109
7.30.3.3	multOp	109
7.30.4	Member Data Documentation	109
7.30.4.1	M	109
7.31	Interchange Class Reference	109
7.31.1	Constructor & Destructor Documentation	114
7.31.1.1	Interchange	114
7.31.1.2	Interchange	114
7.31.1.3	~Interchange	114
7.31.2	Member Function Documentation	114
7.31.2.1	calcSP	114
7.31.2.2	calcula	114
7.31.2.3	clear	114
7.31.2.4	clear	114
7.31.2.5	diff	114
7.31.2.6	diffValues	114
7.31.2.7	fromSubdomains	114
7.31.2.8	genGeom	115
7.31.2.9	genInv	115
7.31.2.10	getCoordNode	115
7.31.2.11	getMaxBdSize	115
7.31.2.12	getND	115
7.31.2.13	getnDim	115
7.31.2.14	getNDuals	115
7.31.2.15	getnOmega	115
7.31.2.16	getNP	115
7.31.2.17	getNPrimals	115
7.31.2.18	getNtype	115
7.31.2.19	getPrimals	115
7.31.2.20	getValue	115

7.31.2.21	getValue	115
7.31.2.22	initialize	115
7.31.2.23	inverse	115
7.31.2.24	inverse	116
7.31.2.25	knownValues	116
7.31.2.26	multOp	116
7.31.2.27	multOp	116
7.31.2.28	pbdValues	116
7.31.2.29	print	116
7.31.2.30	print	116
7.31.2.31	rbdValues	116
7.31.2.32	rhs	116
7.31.2.33	sbdValues	116
7.31.2.34	setNtype	116
7.31.2.35	setPrimals	116
7.31.2.36	setValue	117
7.31.2.37	toSubdomains	117
7.31.3	Member Data Documentation	117
7.31.3.1	Ax	117
7.31.3.2	ax	117
7.31.3.3	Ay	117
7.31.3.4	ay	117
7.31.3.5	Az	117
7.31.3.6	az	117
7.31.3.7	bds	117
7.31.3.8	bdValues	117
7.31.3.9	bx	117
7.31.3.10	by	117
7.31.3.11	bz	117
7.31.3.12	c	117
7.31.3.13	ce	117
7.31.3.14	domain	117
7.31.3.15	f	117
7.31.3.16	fc	117
7.31.3.17	g	117
7.31.3.18	gc	117
7.31.3.19	mesh	117

7.31.3.20 method	117
7.31.3.21 nDim	118
7.31.3.22 nOmega	118
7.31.3.23 omegas	118
7.31.3.24 one	118
7.31.3.25 op	118
7.31.3.26 prim	118
7.31.3.27 primal	118
7.31.3.28 props	118
7.31.3.29 sf	118
7.31.3.30 sg	118
7.31.3.31 swprint	118
7.31.3.32 zero	118
7.32 InterchangeMPI Class Reference	118
7.32.1 Constructor & Destructor Documentation	122
7.32.1.1 InterchangeMPI	122
7.32.2 Member Function Documentation	122
7.32.2.1 calcula	122
7.32.2.2 clear	122
7.32.2.3 diff	122
7.32.2.4 diffValues	122
7.32.2.5 fromSubdomains	122
7.32.2.6 genInv	122
7.32.2.7 getCoordNode	122
7.32.2.8 getInternalBd	122
7.32.2.9 getMaxBdSize	123
7.32.2.10 getNtype	123
7.32.2.11 getPrimals	123
7.32.2.12 getValue	123
7.32.2.13 inverse	123
7.32.2.14 knownValues	123
7.32.2.15 multOp	123
7.32.2.16 print	123
7.32.2.17 print	123
7.32.2.18 rhs	123
7.32.2.19 setNtype	124
7.32.2.20 setPrimals	124

7.32.2.21	toSubdomains	124
7.32.3	Member Data Documentation	124
7.32.3.1	indl	124
7.32.3.2	ME	124
7.32.3.3	msa	124
7.32.3.4	mss	124
7.32.3.5	xnp	124
7.33	InternalBd Class Reference	124
7.33.1	Constructor & Destructor Documentation	126
7.33.1.1	InternalBd	126
7.33.1.2	InternalBd	126
7.33.1.3	~InternalBd	126
7.33.2	Member Function Documentation	126
7.33.2.1	compareTo	126
7.33.2.2	equals	126
7.33.2.3	getval	126
7.33.2.4	rbd	126
7.33.2.5	rcoord	126
7.33.2.6	rdp	126
7.33.2.7	rnode	126
7.33.2.8	rsubd	126
7.33.3	Member Data Documentation	126
7.33.3.1	bd	126
7.33.3.2	coord	126
7.33.3.3	dp	126
7.33.3.4	nDim	126
7.33.3.5	node	126
7.33.3.6	subd	127
7.34	LM1 Class Reference	127
7.34.1	Constructor & Destructor Documentation	130
7.34.1.1	LM1	130
7.34.2	Member Function Documentation	130
7.34.2.1	clean	130
7.34.2.2	dot	130
7.34.2.3	multOp	130
7.34.2.4	rhs	130
7.34.2.5	solve	130

7.34.3	Member Data Documentation	131
7.34.3.1	f	131
7.34.3.2	op	131
7.34.3.3	uf	131
7.34.3.4	up	131
7.35	LM1MPI Class Reference	131
7.35.1	Detailed Description	134
7.35.2	Constructor & Destructor Documentation	134
7.35.2.1	LM1MPI	134
7.35.3	Member Function Documentation	134
7.35.3.1	clean	134
7.35.3.2	iniInterchage	134
7.35.3.3	solvePar	134
7.36	LM2 Class Reference	135
7.36.1	Constructor & Destructor Documentation	137
7.36.1.1	LM2	137
7.36.2	Member Function Documentation	137
7.36.2.1	clean	137
7.36.2.2	dot	137
7.36.2.3	multOp	137
7.36.2.4	rhs	137
7.36.2.5	solve	137
7.36.3	Member Data Documentation	138
7.36.3.1	f	138
7.36.3.2	op	138
7.36.3.3	uf	138
7.36.3.4	ug	138
7.36.3.5	up	138
7.37	LM2MPI Class Reference	138
7.37.1	Detailed Description	141
7.37.2	Constructor & Destructor Documentation	141
7.37.2.1	LM2MPI	141
7.37.3	Member Function Documentation	141
7.37.3.1	clean	141
7.37.3.2	iniInterchage	141
7.37.3.3	solvePar	141
7.38	LookUpFunction Class Reference	142

7.38.1	Member Function Documentation	143
7.38.1.1	getF	143
7.38.2	Member Data Documentation	143
7.38.2.1	ce	143
7.39	Matriz_Base Class Reference	143
7.39.1	Detailed Description	146
7.39.2	Constructor & Destructor Documentation	146
7.39.2.1	Matriz_Base	146
7.39.2.2	~Matriz_Base	147
7.39.3	Member Function Documentation	147
7.39.3.1	asignaNombre	147
7.39.3.2	columnas	147
7.39.3.3	renglones	147
7.39.4	Member Data Documentation	147
7.39.4.1	Ban	147
7.39.4.2	ce	147
7.39.4.3	Col	147
7.39.4.4	Nmb	147
7.39.4.5	Ren	148
7.40	MatrizDispersa Class Reference	148
7.40.1	Detailed Description	151
7.40.2	Constructor & Destructor Documentation	152
7.40.2.1	MatrizDispersa	152
7.40.2.2	MatrizDispersa	152
7.40.2.3	~MatrizDispersa	152
7.40.3	Member Function Documentation	152
7.40.3.1	asigna	152
7.40.3.2	inicializa	153
7.40.3.3	multiplica	153
7.40.3.4	retorna	153
7.40.3.5	retornaNumeroColumna	153
7.40.3.6	retornaNumeroColumnasBanda	153
7.40.3.7	retornaValorColumna	154
7.40.3.8	solicitaMemoria	154
7.40.3.9	tamanoBanda	154
7.40.3.10	visualiza	154
7.40.3.11	visualizaMatricesInternas	155

7.40.4	Member Data Documentation	155
7.40.4.1	J	155
7.40.4.2	M	155
7.41	MF1 Class Reference	155
7.41.1	Constructor & Destructor Documentation	158
7.41.1.1	MF1	158
7.41.2	Member Function Documentation	158
7.41.2.1	clean	158
7.41.2.2	dot	158
7.41.2.3	multOp	158
7.41.2.4	rhs	158
7.41.2.5	solve	158
7.41.3	Member Data Documentation	158
7.41.3.1	op	158
7.42	MF1MPI Class Reference	159
7.42.1	Detailed Description	162
7.42.2	Constructor & Destructor Documentation	162
7.42.2.1	MF1MPI	162
7.42.3	Member Function Documentation	162
7.42.3.1	clean	162
7.42.3.2	iniInterchage	162
7.42.3.3	solvePar	163
7.43	MF2 Class Reference	163
7.43.1	Constructor & Destructor Documentation	166
7.43.1.1	MF2	166
7.43.2	Member Function Documentation	166
7.43.2.1	clean	166
7.43.2.2	dot	166
7.43.2.3	multOp	166
7.43.2.4	rhs	166
7.43.2.5	solve	166
7.43.3	Member Data Documentation	166
7.43.3.1	op	166
7.43.3.2	up	167
7.44	MF2MPI Class Reference	167
7.44.1	Detailed Description	170
7.44.2	Constructor & Destructor Documentation	170

7.44.2.1	MF2MPI	170
7.44.3	Member Function Documentation	170
7.44.3.1	clean	170
7.44.3.2	iniInterchage	170
7.44.3.3	solvePar	170
7.45	MultBandSym Class Reference	171
7.45.1	Constructor & Destructor Documentation	172
7.45.1.1	MultBandSym	172
7.45.2	Member Function Documentation	172
7.45.2.1	getSize	173
7.45.2.2	multOp	173
7.45.3	Member Data Documentation	173
7.45.3.1	AK	173
7.45.3.2	bw	173
7.45.3.3	n	173
7.46	MultOp Class Reference	173
7.46.1	Member Function Documentation	175
7.46.1.1	getSize	175
7.46.1.2	multOp	175
7.47	NoPrimal Class Reference	175
7.47.1	Constructor & Destructor Documentation	177
7.47.1.1	NoPrimal	177
7.47.2	Member Function Documentation	177
7.47.2.1	isPrimal	177
7.47.3	Member Data Documentation	177
7.47.3.1	name	177
7.48	NSfExpXY Class Reference	177
7.48.1	Constructor & Destructor Documentation	180
7.48.1.1	NSfExpXY	180
7.48.2	Member Function Documentation	180
7.48.2.1	eval	180
7.48.2.2	getVar	180
7.48.2.3	setVar	180
7.48.3	Member Data Documentation	180
7.48.3.1	var	180
7.49	NSfExpXYZ Class Reference	180
7.49.1	Constructor & Destructor Documentation	183

7.49.1.1	NSfExpXYZ	183
7.49.2	Member Function Documentation	183
7.49.2.1	eval	183
7.49.2.2	getVar	183
7.49.2.3	setVar	183
7.49.3	Member Data Documentation	183
7.49.3.1	var	183
7.50	PLM1 Class Reference	183
7.50.1	Constructor & Destructor Documentation	186
7.50.1.1	PLM1	186
7.50.2	Member Function Documentation	186
7.50.2.1	clean	186
7.50.2.2	dot	186
7.50.2.3	multOp	186
7.50.2.4	rhs	186
7.50.2.5	solve	186
7.50.3	Member Data Documentation	187
7.50.3.1	f	187
7.50.3.2	op	187
7.50.3.3	uf	187
7.50.3.4	up	187
7.51	PLM1MPI Class Reference	187
7.51.1	Detailed Description	190
7.51.2	Constructor & Destructor Documentation	190
7.51.2.1	PLM1MPI	190
7.51.3	Member Function Documentation	190
7.51.3.1	clean	190
7.51.3.2	iniInterchage	190
7.51.3.3	solvePar	190
7.52	PLM2 Class Reference	191
7.52.1	Constructor & Destructor Documentation	193
7.52.1.1	PLM2	193
7.52.2	Member Function Documentation	193
7.52.2.1	clean	193
7.52.2.2	dot	193
7.52.2.3	multOp	193
7.52.2.4	rhs	193

7.52.2.5	solve	193
7.52.3	Member Data Documentation	194
7.52.3.1	f	194
7.52.3.2	op	194
7.52.3.3	uf	194
7.52.3.4	ug	194
7.52.3.5	up	194
7.53	PLM2MPI Class Reference	194
7.53.1	Detailed Description	197
7.53.2	Constructor & Destructor Documentation	197
7.53.2.1	PLM2MPI	197
7.53.3	Member Function Documentation	197
7.53.3.1	clean	197
7.53.3.2	iniInterchage	197
7.53.3.3	solvePar	197
7.54	PMF1 Class Reference	198
7.54.1	Constructor & Destructor Documentation	200
7.54.1.1	PMF1	200
7.54.2	Member Function Documentation	200
7.54.2.1	clean	200
7.54.2.2	dot	200
7.54.2.3	multOp	200
7.54.2.4	rhs	200
7.54.2.5	solve	200
7.54.3	Member Data Documentation	200
7.54.3.1	op	200
7.55	PMF1MPI Class Reference	201
7.55.1	Detailed Description	204
7.55.2	Constructor & Destructor Documentation	204
7.55.2.1	PMF1MPI	204
7.55.3	Member Function Documentation	204
7.55.3.1	clean	204
7.55.3.2	iniInterchage	204
7.55.3.3	solvePar	205
7.56	PMF2 Class Reference	205
7.56.1	Constructor & Destructor Documentation	208
7.56.1.1	PMF2	208

7.56.2	Member Function Documentation	208
7.56.2.1	clean	208
7.56.2.2	dot	208
7.56.2.3	multOp	208
7.56.2.4	rhs	208
7.56.2.5	solve	208
7.56.3	Member Data Documentation	208
7.56.3.1	op	208
7.56.3.2	uf	209
7.56.3.3	up	209
7.57	PMF2MPI Class Reference	209
7.57.1	Detailed Description	212
7.57.2	Constructor & Destructor Documentation	212
7.57.2.1	PMF2MPI	212
7.57.3	Member Function Documentation	212
7.57.3.1	clean	212
7.57.3.2	iniInterchage	212
7.57.3.3	solvePar	212
7.58	Primal Class Reference	213
7.58.1	Constructor & Destructor Documentation	215
7.58.1.1	Primal	215
7.58.1.2	~Primal	215
7.58.2	Member Function Documentation	215
7.58.2.1	isPrimal	215
7.58.3	Member Data Documentation	215
7.58.3.1	EDGE	215
7.58.3.2	FACE	215
7.58.3.3	INTBD	215
7.58.3.4	INTERIOR	215
7.58.3.5	KNOWN	215
7.58.3.6	PRIMAL	215
7.58.3.7	VERTEX	215
7.59	PropDef Class Reference	215
7.59.1	Constructor & Destructor Documentation	218
7.59.1.1	PropDef	218
7.59.1.2	PropDef	218
7.59.1.3	PropDef	218

7.59.2	Member Function Documentation	218
7.59.2.1	getDouble	218
7.59.2.2	getDouble	218
7.59.2.3	getInt	218
7.59.2.4	getInt	218
7.59.2.5	getString	218
7.59.2.6	getString	218
7.59.2.7	parse	218
7.59.2.8	parse	218
7.60	Properties Class Reference	218
7.60.1	Constructor & Destructor Documentation	221
7.60.1.1	Properties	221
7.60.2	Member Function Documentation	221
7.60.2.1	getProperty	221
7.60.2.2	getProperty	221
7.60.2.3	list	221
7.60.2.4	load	221
7.60.2.5	setProperty	221
7.60.3	Member Data Documentation	221
7.60.3.1	ce	221
7.60.3.2	keys	221
7.60.3.3	vals	221
7.61	RectSub Class Reference	221
7.61.1	Constructor & Destructor Documentation	225
7.61.1.1	RectSub	225
7.61.1.2	~RectSub	225
7.61.2	Member Function Documentation	225
7.61.2.1	addProjNs	225
7.61.2.2	addProjNs	225
7.61.2.3	clear	225
7.61.2.4	diff	225
7.61.2.5	diffValues	225
7.61.2.6	genCoef	225
7.61.2.7	genCoefVar	225
7.61.2.8	genInv	225
7.61.2.9	genInverse	225
7.61.2.10	genNcoord	225

7.61.2.11	genNtype	225
7.61.2.12	getBdSize	225
7.61.2.13	getCoord	225
7.61.2.14	getCoordNode	225
7.61.2.15	getInternalBd	225
7.61.2.16	getInv	225
7.61.2.17	getNP	225
7.61.2.18	getNtype	225
7.61.2.19	getPrimals	225
7.61.2.20	getValue	225
7.61.2.21	getValues	225
7.61.2.22	inverse	225
7.61.2.23	isDual	226
7.61.2.24	isFloat	226
7.61.2.25	isIntBd	226
7.61.2.26	isInterior	226
7.61.2.27	isInterior	226
7.61.2.28	isKnown	226
7.61.2.29	isKnown	226
7.61.2.30	isPrimal	226
7.61.2.31	isVertex	226
7.61.2.32	knownValues	226
7.61.2.33	multOp	226
7.61.2.34	nodeType	226
7.61.2.35	print	226
7.61.2.36	print	226
7.61.2.37	printMat	226
7.61.2.38	printMult	226
7.61.2.39	rhs	226
7.61.2.40	setNtype	226
7.61.2.41	setPrimals	226
7.61.2.42	setValue	226
7.61.2.43	setValues	226
7.61.3	Member Data Documentation	226
7.61.3.1	bdMap	226
7.61.3.2	bFloat	226
7.61.3.3	bsym	226

7.61.3.4 Ca	226
7.61.3.5 Cb	227
7.61.3.6 Cc	227
7.61.3.7 ce	227
7.61.3.8 coef	227
7.61.3.9 coord	227
7.61.3.10 coordN	227
7.61.3.11 domain	227
7.61.3.12 DUAL	227
7.61.3.13 EDGE	227
7.61.3.14 FACE	227
7.61.3.15 h	227
7.61.3.16 hfac	227
7.61.3.17 id	227
7.61.3.18 INTBD	227
7.61.3.19 INTERIOR	227
7.61.3.20 inv	227
7.61.3.21 KNOWN	227
7.61.3.22 M	227
7.61.3.23 M1	227
7.61.3.24 mapFull	227
7.61.3.25 mapInt	227
7.61.3.26 mesh	227
7.61.3.27 N	227
7.61.3.28 nBd	227
7.61.3.29 nDim	227
7.61.3.30 nFull	228
7.61.3.31 nInt	228
7.61.3.32 np	228
7.61.3.33 ntype	228
7.61.3.34 op	228
7.61.3.35 PRIMAL	228
7.61.3.36 scr	228
7.61.3.37 VERTEX	228
7.61.3.38 X	228
7.61.3.39 x	228
7.61.3.40 Y	228

7.62 SfExpXYZ Class Reference	228
7.62.1 Constructor & Destructor Documentation	231
7.62.1.1 SfExpXYZ	231
7.62.2 Member Function Documentation	231
7.62.2.1 eval	231
7.62.2.2 getVar	231
7.62.2.3 setVar	231
7.62.3 Member Data Documentation	231
7.62.3.1 var	231
7.63 SinPinxSinPiny Class Reference	231
7.63.1 Constructor & Destructor Documentation	234
7.63.1.1 SinPinxSinPiny	234
7.63.2 Member Function Documentation	234
7.63.2.1 eval	234
7.63.2.2 getVar	234
7.63.2.3 setVar	234
7.63.3 Member Data Documentation	234
7.63.3.1 n	234
7.63.3.2 var	234
7.64 SinPinxSinPinySinPinz Class Reference	234
7.64.1 Constructor & Destructor Documentation	237
7.64.1.1 SinPinxSinPinySinPinz	237
7.64.2 Member Function Documentation	237
7.64.2.1 eval	237
7.64.2.2 getVar	237
7.64.2.3 setVar	237
7.64.3 Member Data Documentation	237
7.64.3.1 n	237
7.64.3.2 var	237
7.65 SinPix Class Reference	237
7.65.1 Constructor & Destructor Documentation	240
7.65.1.1 SinPix	240
7.65.2 Member Function Documentation	240
7.65.2.1 eval	240
7.65.2.2 getVar	240
7.65.2.3 setVar	240
7.65.3 Member Data Documentation	240

7.65.3.1	var	240
7.66	SinPixCosPiy Class Reference	240
7.66.1	Constructor & Destructor Documentation	243
7.66.1.1	SinPixCosPiy	243
7.66.2	Member Function Documentation	243
7.66.2.1	eval	243
7.66.2.2	getVar	243
7.66.2.3	setVar	243
7.66.3	Member Data Documentation	243
7.66.3.1	var	243
7.67	SinPixSinPiy Class Reference	243
7.67.1	Constructor & Destructor Documentation	246
7.67.1.1	SinPixSinPiy	246
7.67.2	Member Function Documentation	246
7.67.2.1	eval	246
7.67.2.2	getVar	246
7.67.2.3	setVar	246
7.67.3	Member Data Documentation	246
7.67.3.1	var	246
7.68	SinPiXSinPiYSinPiZ Class Reference	246
7.68.1	Constructor & Destructor Documentation	249
7.68.1.1	SinPiXSinPiYSinPiZ	249
7.68.2	Member Function Documentation	249
7.68.2.1	eval	249
7.68.2.2	getVar	249
7.68.2.3	setVar	249
7.68.3	Member Data Documentation	249
7.68.3.1	var	249
7.69	Solvable Class Reference	249
7.69.1	Constructor & Destructor Documentation	251
7.69.1.1	Solvable	251
7.69.1.2	~Solvable	251
7.69.2	Member Function Documentation	251
7.69.2.1	clean	252
7.69.2.2	getIter	252
7.69.2.3	getName	252
7.69.2.4	solve	252

7.69.3	Member Data Documentation	252
7.69.3.1	name	252
7.70	VertEdgePrimal Class Reference	252
7.70.1	Constructor & Destructor Documentation	255
7.70.1.1	VertEdgePrimal	255
7.70.2	Member Function Documentation	255
7.70.2.1	isPrimal	255
7.70.3	Member Data Documentation	255
7.70.3.1	name	255
7.71	VertPrimal Class Reference	255
7.71.1	Constructor & Destructor Documentation	258
7.71.1.1	VertPrimal	258
7.71.2	Member Function Documentation	258
7.71.2.1	isPrimal	258
7.71.3	Member Data Documentation	258
7.71.3.1	name	258
8	File Documentation	259
8.1	AllPrimal.hpp File Reference	259
8.2	BandCholesky.cpp File Reference	260
8.3	BandCholesky.hpp File Reference	260
8.4	BandSolve.cpp File Reference	262
8.5	BandSolve.hpp File Reference	262
8.6	BdNode.hpp File Reference	264
8.7	CGM.cpp File Reference	264
8.8	CGM.hpp File Reference	265
8.9	Constant.hpp File Reference	265
8.10	CreateBdNodes.hpp File Reference	266
8.11	Definiciones.hpp File Reference	267
8.11.1	Macro Definition Documentation	268
8.11.1.1	__Double__	268
8.11.1.2	COEFICIENTES_CONSTANTES	268
8.11.1.3	DIM_VECTOR	268
8.11.1.4	EPS_EQUAL	268
8.11.1.5	EPSILON	268
8.11.1.6	EPSILON_LOCAL	268
8.11.1.7	NMAXITER	268

8.11.1.8	NMAXITER_LOCAL	269
8.11.1.9	RESIDUAL	269
8.11.2	Typedef Documentation	269
8.11.2.1	ldouble	269
8.12	Disc.hpp File Reference	269
8.13	Disc13.hpp File Reference	270
8.14	Disc14.hpp File Reference	271
8.15	Disc15.hpp File Reference	272
8.16	DotProd.hpp File Reference	273
8.17	DPMainMPI.cpp File Reference	274
8.18	DPMainMPI.hpp File Reference	274
8.19	DPMethod.cpp File Reference	275
8.20	DPMethod.hpp File Reference	275
8.21	DQGMRES.cpp File Reference	276
8.22	DQGMRES.hpp File Reference	277
8.23	DualPrimal.cpp File Reference	277
8.24	DualPrimal.hpp File Reference	278
8.25	EllipOp.hpp File Reference	279
8.26	ErrorControl.cpp File Reference	279
8.27	ErrorControl.hpp File Reference	280
8.28	EsquemaMEMPI.cpp File Reference	281
8.29	EsquemaMEMPI.hpp File Reference	282
8.30	ExpVXY.hpp File Reference	282
8.31	ExpVXYZ.hpp File Reference	283
8.32	ExpX.hpp File Reference	285
8.33	ExpXY.hpp File Reference	286
8.34	ExpXYZ.hpp File Reference	287
8.35	fExpXY.hpp File Reference	288
8.36	FunctionV.hpp File Reference	289
8.37	FunctionV1.hpp File Reference	289
8.38	HeapSort.hpp File Reference	290
8.39	ICGM.hpp File Reference	291
8.40	IDQGMRES.hpp File Reference	292
8.41	Interchange.cpp File Reference	293
8.42	Interchange.hpp File Reference	294
8.43	InterchangeMPI.cpp File Reference	295
8.44	InterchangeMPI.hpp File Reference	295

8.45	InternalBd.hpp File Reference	296
8.46	LM1.cpp File Reference	296
8.47	LM1.hpp File Reference	297
8.48	LM1MPI.hpp File Reference	298
8.49	LM2.cpp File Reference	298
8.50	LM2.hpp File Reference	298
8.51	LM2MPI.hpp File Reference	299
8.52	LookUpFunction.cpp File Reference	300
8.53	LookUpFunction.hpp File Reference	301
8.54	main.hpp File Reference	301
8.55	Matriz_Base.hpp File Reference	301
8.56	MatrizDispersa.cpp File Reference	302
8.57	MatrizDispersa.hpp File Reference	303
8.58	MF1.cpp File Reference	304
8.59	MF1.hpp File Reference	305
8.60	MF1MPI.hpp File Reference	306
8.61	MF2.cpp File Reference	306
8.62	MF2.hpp File Reference	306
8.63	MF2MPI.hpp File Reference	307
8.64	MultBandSym.cpp File Reference	308
8.65	MultBandSym.hpp File Reference	309
8.66	MultOp.hpp File Reference	309
8.67	NoPrimal.hpp File Reference	310
8.68	NSfExpXY.hpp File Reference	311
8.69	NSfExpXYZ.hpp File Reference	312
8.70	PLM1.cpp File Reference	313
8.71	PLM1.hpp File Reference	313
8.72	PLM1MPI.hpp File Reference	314
8.73	PLM2.cpp File Reference	315
8.74	PLM2.hpp File Reference	315
8.75	PLM2MPI.hpp File Reference	316
8.76	PMF1.cpp File Reference	316
8.77	PMF1.hpp File Reference	317
8.78	PMF1MPI.hpp File Reference	318
8.79	PMF2.cpp File Reference	318
8.80	PMF2.hpp File Reference	318
8.81	PMF2MPI.hpp File Reference	319

8.82	Primal.hpp File Reference	320
8.83	PropDef.cpp File Reference	320
8.84	PropDef.hpp File Reference	321
8.85	Properties.cpp File Reference	322
8.86	Properties.hpp File Reference	322
8.87	RectSub.cpp File Reference	323
8.88	RectSub.hpp File Reference	324
8.89	SfExpXYZ.hpp File Reference	325
8.90	SinPinxSinPiny.hpp File Reference	326
8.91	SinPinxSinPinySinPinz.hpp File Reference	327
8.92	SinPix.hpp File Reference	328
8.93	SinPixCosPiy.hpp File Reference	329
8.94	SinPixSinPiy.hpp File Reference	330
8.95	SinPiXSinPiYSinPiZ.hpp File Reference	331
8.96	Solvable.hpp File Reference	332
8.97	VertEdgePrimal.hpp File Reference	332
8.98	VertPrimal.hpp File Reference	333
9	Example Documentation	335
9.1	EjemploMatrizDispersa.cpp	335
9.2	ExampleErrorControl.cpp	335

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 tamaño de banda

Multiplicación 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

Class Index

4.1 Class Hierarchy

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

BdNode	25
CreateBdNodes	34
DotProd	48
DPMMethod	55
LM1	127
LM1MPI	131
LM2	135
LM2MPI	138
MF1	155
MF1MPI	159
MF2	163
MF2MPI	167
PLM1	183
PLM1MPI	187
PLM2	191
PLM2MPI	194
PMF1	198
PMF1MPI	201
PMF2	205
PMF2MPI	209
ICGM	101
DualPrimal	64
EllipOp	66
ErrorControl	69
EsquemaMEMPI	73
DPMMainMPI	50
LM1MPI	131
LM2MPI	138
MF1MPI	159
MF2MPI	167
PLM1MPI	187
PLM2MPI	194

PMF1MPI	201
PMF2MPI	209
FunctionV	96
FunctionV1	98
Constant	32
Disc	37
Disc13	39
Disc14	42
Disc15	45
ExpVXY	78
ExpVXYZ	81
ExpX	84
ExpXY	87
ExpXYZ	90
fExpXY	93
NSfExpXY	177
NSfExpXYZ	180
SfExpXYZ	228
SinPinxSinPiny	231
SinPinxSinPinySinPinz	234
SinPix	237
SinPixCosPiy	240
SinPixSinPiy	243
SinPiXSinPiYSinPiZ	246
HeapSort	100
Interchange	109
InterchangeMPI	118
InternalBd	124
LookUpFunction	142
Matriz_Base	143
MatrizDispersa	148
MultOp	173
DPMethod	55
ICGM	101
IDQGMRES	105
MultBandSym	171
Primal	213
AllPrimal	15
NoPrimal	175
VertEdgePrimal	252
VertPrimal	255
Properties	218
PropDef	215
RectSub	221
Solvable	249
BandCholesky	17
BandSolve	21
CGM	27
ICGM	101
DQGMRES	59
IDQGMRES	105

Chapter 5

Class Index

5.1 Class List

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

AllPrimal	15
BandCholesky	17
BandSolve	21
BdNode	25
CGM	27
Constant	32
CreateBdNodes	34
Disc	37
Disc13	39
Disc14	42
Disc15	45
DotProd	48
DPMainMPI	
Clase base para definir a los metodos DVS-DDM	50
DPMethod	55
DQGMRES	59
DualPrimal	64
EllipOp	66
ErrorControl	
Error Control	69
EsquemaMEMPI	
Clase base para definir el Esquema Maestro-Esclavo en MPI	73
ExpVXY	78
ExpVXYZ	81
ExpX	84
ExpXY	87
ExpXYZ	90
fExpXY	93
FunctionV	96
FunctionV1	98
HeapSort	100
ICGM	
Clase para implementar CGM con matrices bandadas o dispersas	101

IDQGMRES	
Clase para implementar DQGMRES con matrices bandadas o dispersas	105
Interchange	109
InterchangeMPI	118
InternalBd	124
LM1	127
LM1MPI	
Clase para definir el metodo LM-1 de DVS-DDM	131
LM2	135
LM2MPI	
Clase para definir el metodo LM-2 de DVS-DDM	138
LookUpFunction	142
Matriz_Base	
Clase base para el trabajar con matrices	143
MatrizDispersa	148
MF1	155
MF1MPI	
Clase para definir el metodo MF-1 de DVS-DDM	159
MF2	163
MF2MPI	
Clase para definir el metodo MF-2 de DVS-DDM	167
MultBandSym	171
MultOp	173
NoPrimal	175
NSfExpXY	177
NSfExpXYZ	180
PLM1	183
PLM1MPI	
Clase para definir el metodo PLM-1 de DVS-DDM	187
PLM2	191
PLM2MPI	
Clase para definir el metodo MF-1 de DVS-DDM	194
PMF1	198
PMF1MPI	
Clase para definir el metodo PMF-1 de DVS-DDM	201
PMF2	205
PMF2MPI	
Clase para definir el metodo PMF-2 de DVS-DDM	209
Primal	213
PropDef	215
Properties	218
RectSub	221
SfExpXYZ	228
SinPinxSinPiny	231
SinPinxSinPinySinPinz	234
SinPix	237
SinPixCosPiy	240
SinPixSinPiy	243
SinPiXSinPiYSinPiZ	246
Solvable	249
VertEdgePrimal	252
VertPrimal	255

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

AllPrimal.hpp	259
BandCholesky.cpp	260
BandCholesky.hpp	260
BandSolve.cpp	262
BandSolve.hpp	262
BdNode.hpp	264
CGM.cpp	264
CGM.hpp	265
Constant.hpp	265
CreateBdNodes.hpp	266
Definiciones.hpp	267
Disc.hpp	269
Disc13.hpp	270
Disc14.hpp	271
Disc15.hpp	272
DotProd.hpp	273
DPMainMPI.cpp	274
DPMainMPI.hpp	274
DPMethod.cpp	275
DPMethod.hpp	275
DQGMRES.cpp	276
DQGMRES.hpp	277
DualPrimal.cpp	277
DualPrimal.hpp	278
EllipOp.hpp	279
ErrorControl.cpp	279
ErrorControl.hpp	280
EsquemaMEMPI.cpp	281
EsquemaMEMPI.hpp	282
ExpVXY.hpp	282
ExpVXYZ.hpp	283
ExpX.hpp	285
ExpXY.hpp	286
ExpXYZ.hpp	287

fExpXY.hpp	288
FunctionV.hpp	289
FunctionV1.hpp	289
HeapSort.hpp	290
ICGM.hpp	291
IDQGMRES.hpp	292
Interchange.cpp	293
Interchange.hpp	294
InterchangeMPI.cpp	295
InterchangeMPI.hpp	295
InternalBd.hpp	296
LM1.cpp	296
LM1.hpp	297
LM1MPI.hpp	298
LM2.cpp	298
LM2.hpp	298
LM2MPI.hpp	299
LookUpFunction.cpp	300
LookUpFunction.hpp	301
main.hpp	301
Matriz_Base.hpp	301
MatrizDispersa.cpp	302
MatrizDispersa.hpp	303
MF1.cpp	304
MF1.hpp	305
MF1MPI.hpp	306
MF2.cpp	306
MF2.hpp	306
MF2MPI.hpp	307
MultBandSym.cpp	308
MultBandSym.hpp	309
MultOp.hpp	309
NoPrimal.hpp	310
NSfExpXY.hpp	311
NSfExpXYZ.hpp	312
PLM1.cpp	313
PLM1.hpp	313
PLM1MPI.hpp	314
PLM2.cpp	315
PLM2.hpp	315
PLM2MPI.hpp	316
PMF1.cpp	316
PMF1.hpp	317
PMF1MPI.hpp	318
PMF2.cpp	318
PMF2.hpp	318
PMF2MPI.hpp	319
Primal.hpp	320
PropDef.cpp	320
PropDef.hpp	321
Properties.cpp	322
Properties.hpp	322
RectSub.cpp	323
RectSub.hpp	324

SfExpXYZ.hpp	325
SinPinxSinPiny.hpp	326
SinPinxSinPinySinPinz.hpp	327
SinPix.hpp	328
SinPixCosPiy.hpp	329
SinPixSinPiy.hpp	330
SinPiXSinPiYSinPiZ.hpp	331
Solvable.hpp	332
VertEdgePrimal.hpp	332
VertPrimal.hpp	333

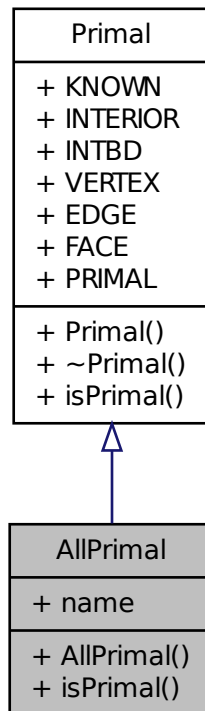
Chapter 7

Class Documentation

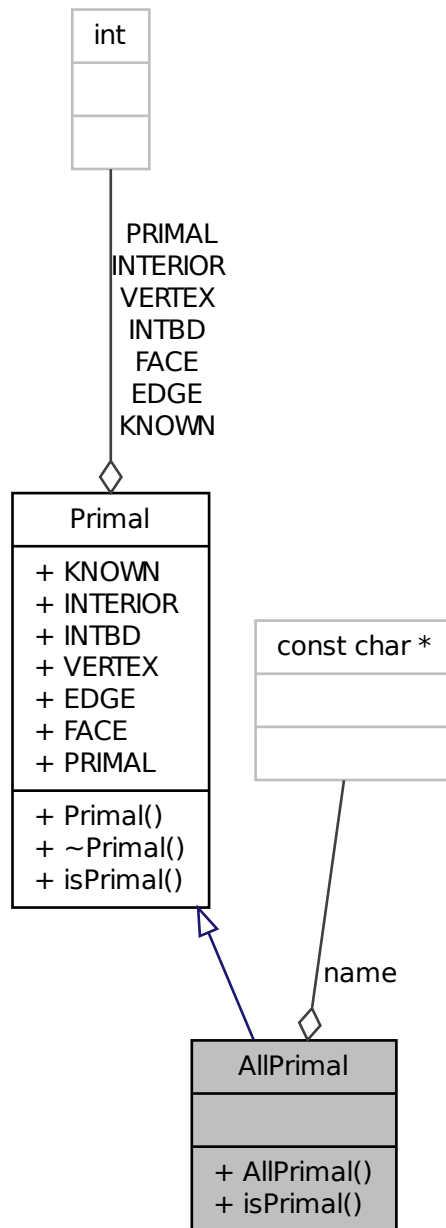
7.1 AllPrimal Class Reference

```
#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`

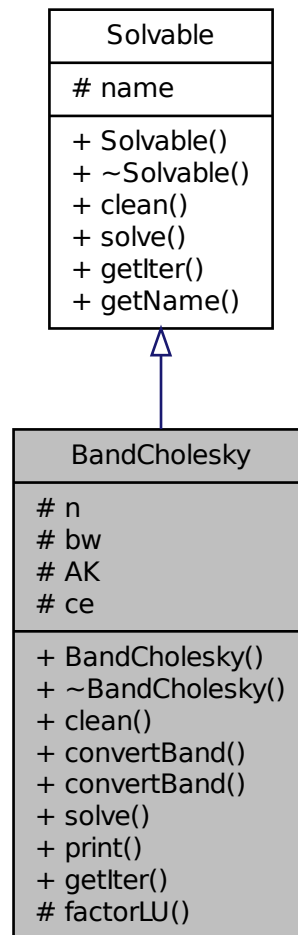
The documentation for this class was generated from the following file:

- [AllPrimal.hpp](#)

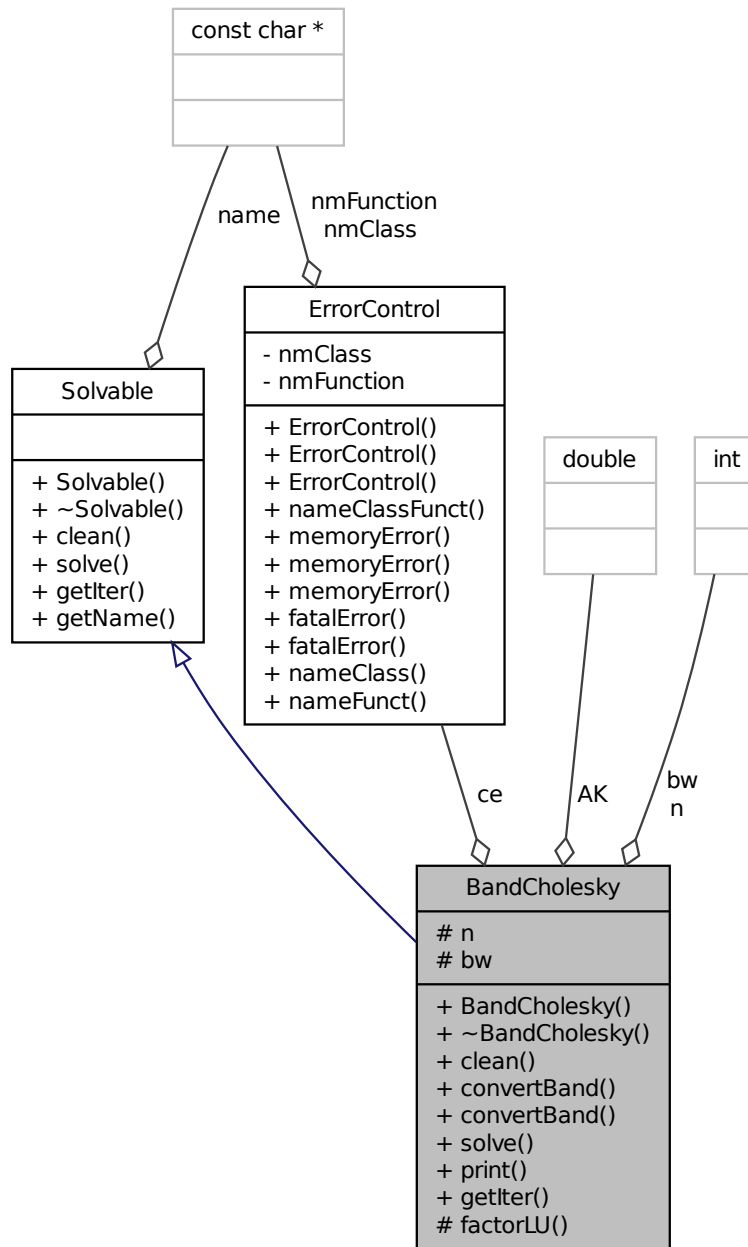
7.2 BandCholesky Class Reference

```
#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]

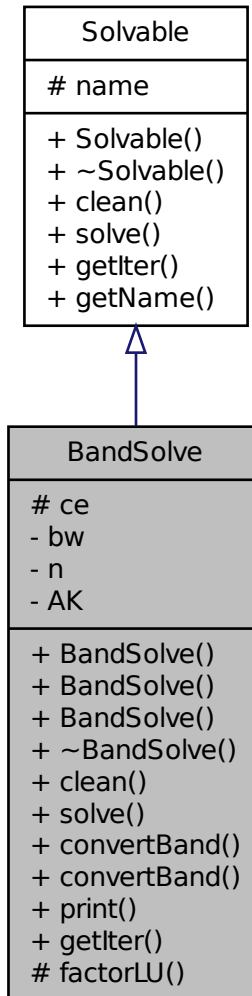
The documentation for this class was generated from the following files:

- [BandCholesky.hpp](#)
- [BandCholesky.cpp](#)

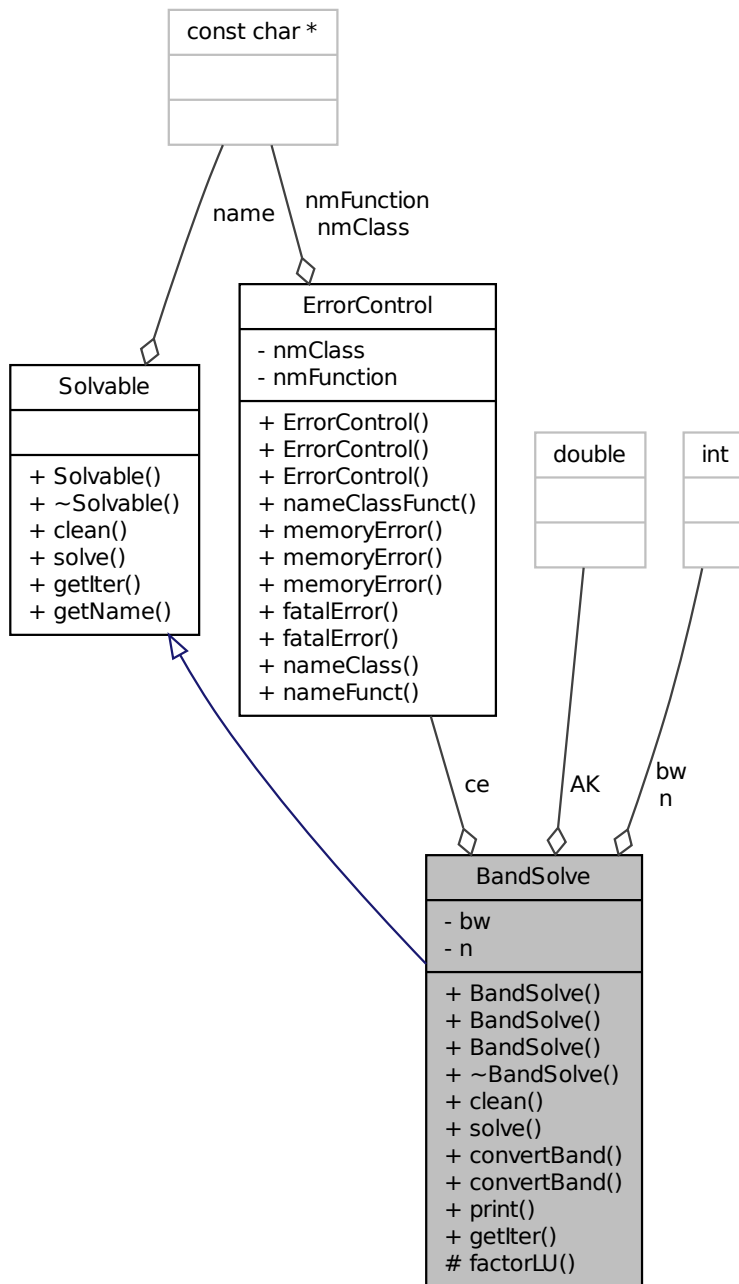
7.3 BandSolve Class Reference

```
#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 (Idouble * x, Idouble * y) [virtual]

Implements [Solvable](#).

7.3.3 Member Data Documentation

7.3.3.1 Idouble** 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]

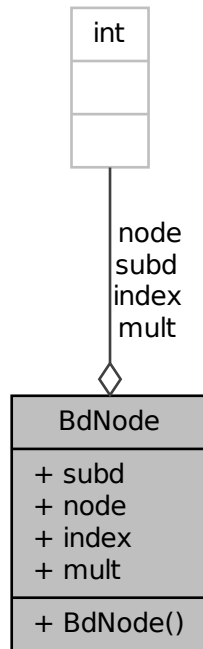
The documentation for this class was generated from the following files:

- [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](#)

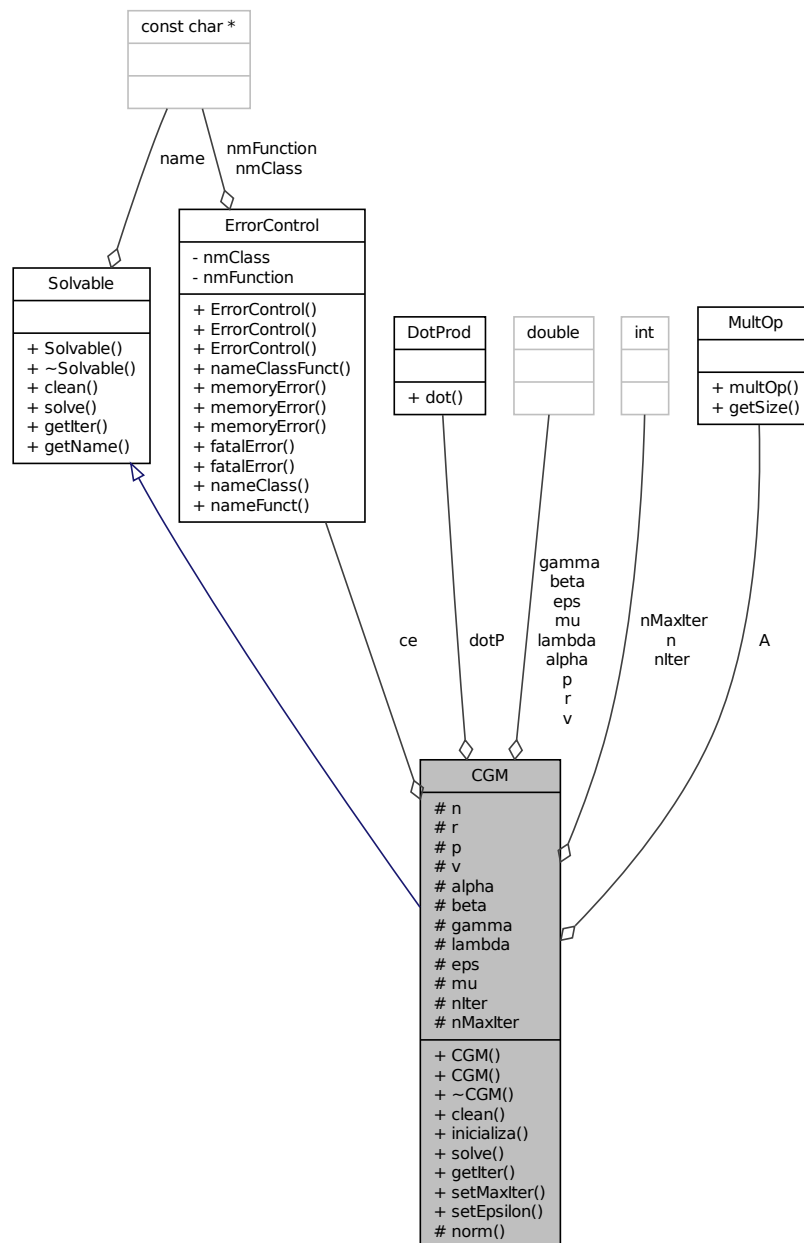
7.5 CGM Class Reference

```
#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 `Idouble CGM::norm (Idouble * x)` [protected]
- 7.5.2.5 `void CGM::setEpsilon (Idouble ep)` [inline]
- 7.5.2.6 `void CGM::setMaxIter (int nmi)` [inline]
- 7.5.2.7 `void CGM::solve (Idouble * u, Idouble * 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]

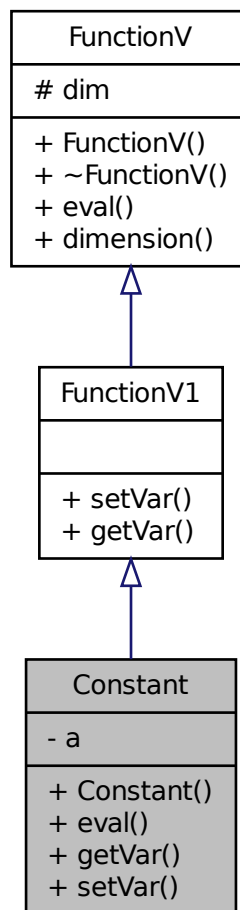
The documentation for this class was generated from the following files:

- [CGM.hpp](#)
- [CGM.cpp](#)

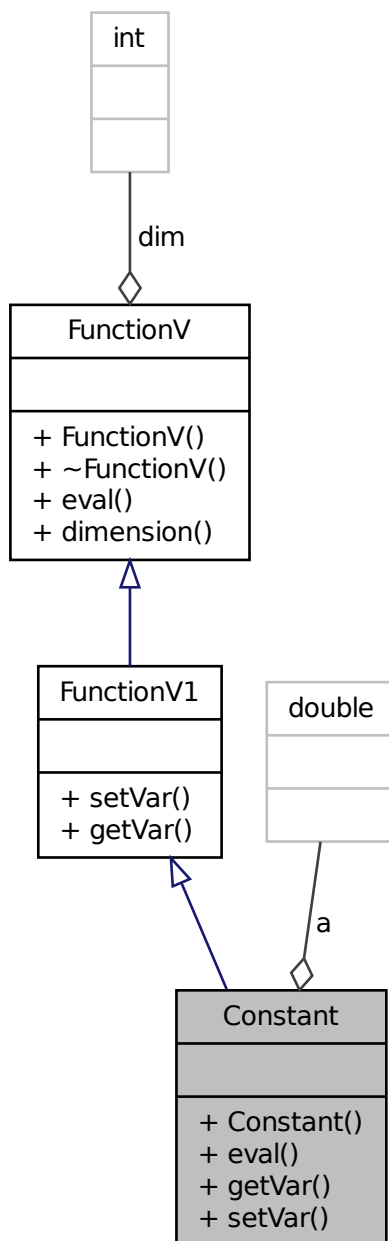
7.6 Constant Class Reference

```
#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](#)]

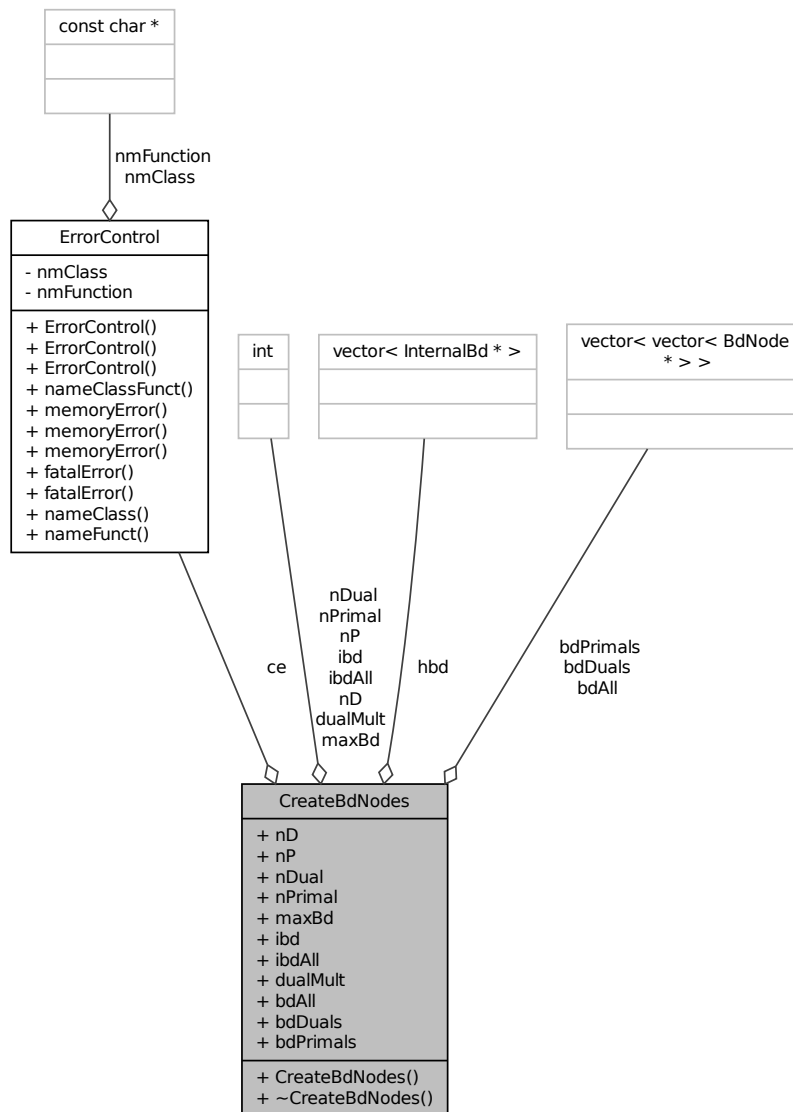
The documentation for this class was generated from the following file:

- [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

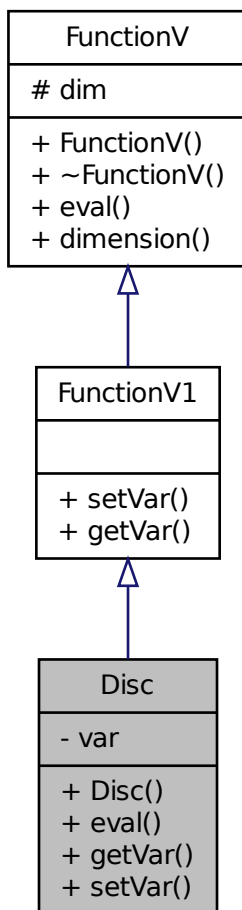
The documentation for this class was generated from the following file:

- [CreateBdNodes.hpp](#)

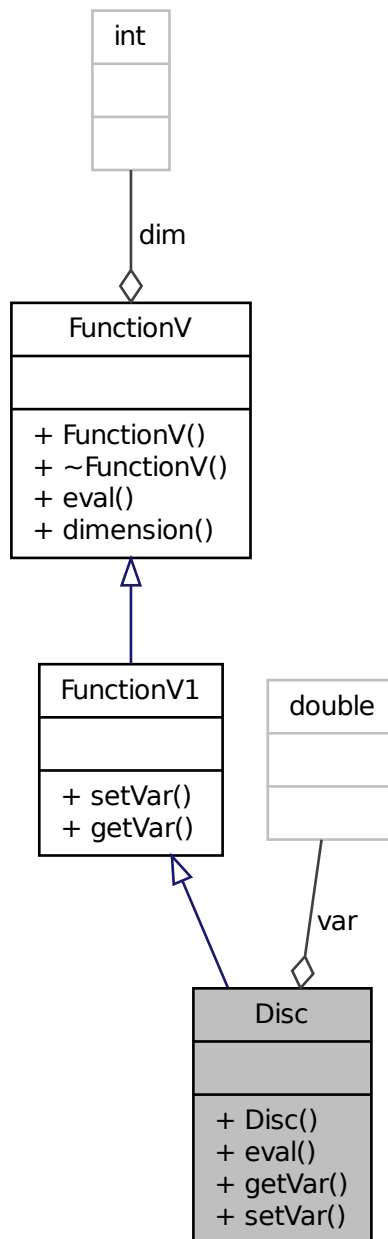
7.8 Disc Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.8.1 Constructor & Destructor Documentation

7.8.1.1 `Disc::Disc (ldouble b)` [`inline`]

7.8.2 Member Function Documentation

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

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

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

Implements [FunctionV1](#).

7.8.3 Member Data Documentation

7.8.3.1 `ldouble Disc::var` [`private`]

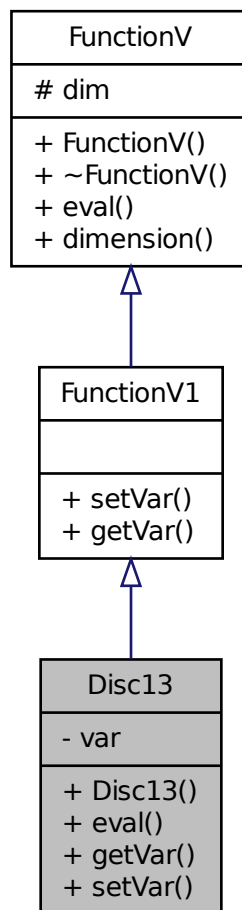
The documentation for this class was generated from the following file:

- [Disc.hpp](#)

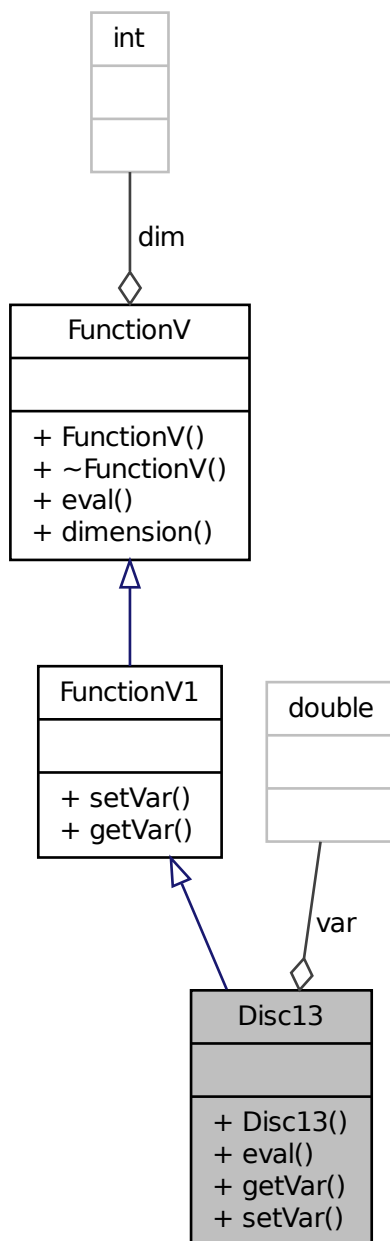
7.9 Disc13 Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.9.1 Constructor & Destructor Documentation

7.9.1.1 `Disc13::Disc13(ldouble b)` [`inline`]

7.9.2 Member Function Documentation

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

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

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

Implements [FunctionV1](#).

7.9.3 Member Data Documentation

7.9.3.1 `ldouble Disc13::var` [`private`]

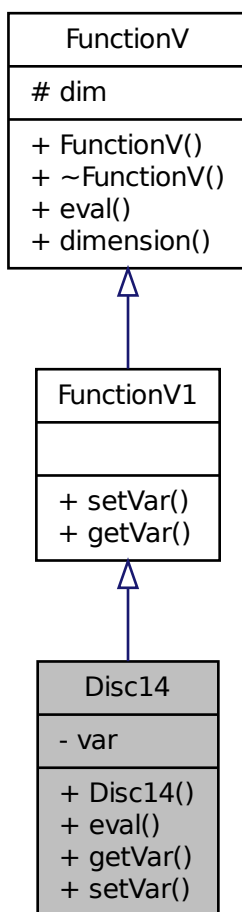
The documentation for this class was generated from the following file:

- [Disc13.hpp](#)

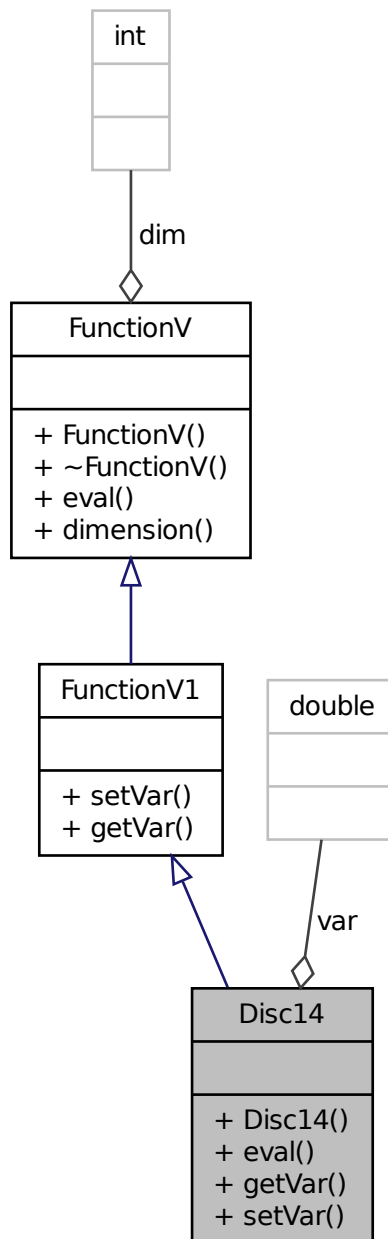
7.10 Disc14 Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.10.1 Constructor & Destructor Documentation

7.10.1.1 `Disc14::Disc14 (ldouble b)` [`inline`]

7.10.2 Member Function Documentation

7.10.2.1 `ldouble Disc14::eval (int d, ldouble * x)` [`inline`],[`virtual`]

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

7.10.2.3 `void Disc14::setVar (ldouble b)` [`inline`],[`virtual`]

Implements [FunctionV1](#).

7.10.3 Member Data Documentation

7.10.3.1 `ldouble Disc14::var` [`private`]

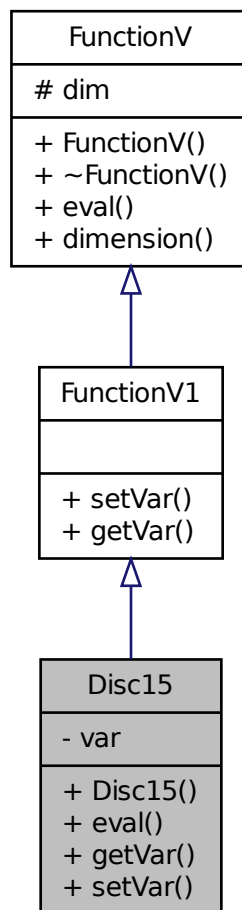
The documentation for this class was generated from the following file:

- [Disc14.hpp](#)

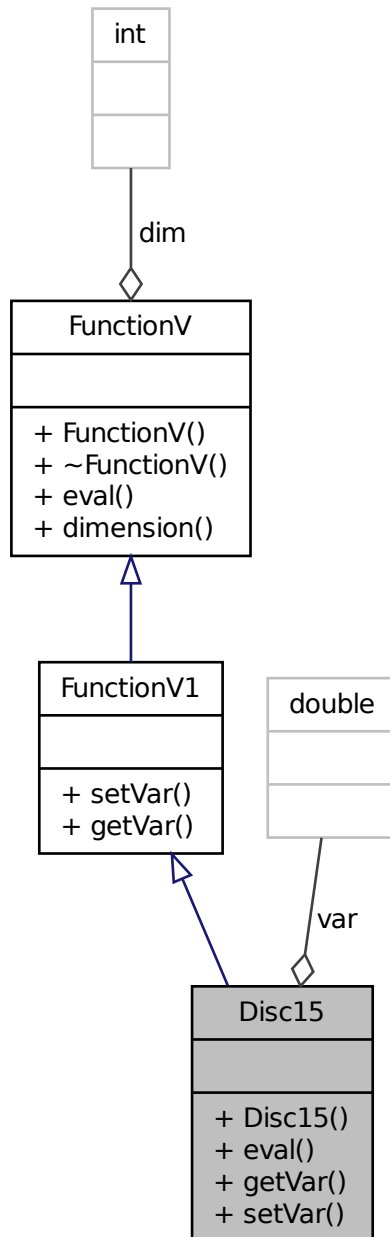
7.11 Disc15 Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.11.1 Constructor & Destructor Documentation

7.11.1.1 `Disc15::Disc15 (ldouble b)` [[inline](#)]

7.11.2 Member Function Documentation

7.11.2.1 `ldouble Disc15::eval (int d, ldouble * x)` [[inline](#)],[[virtual](#)]

Implements [FunctionV](#).

7.11.2.2 `ldouble Disc15::getVar (void)` [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.11.2.3 `void Disc15::setVar (ldouble b)` [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.11.3 Member Data Documentation

7.11.3.1 `ldouble Disc15::var` [[private](#)]

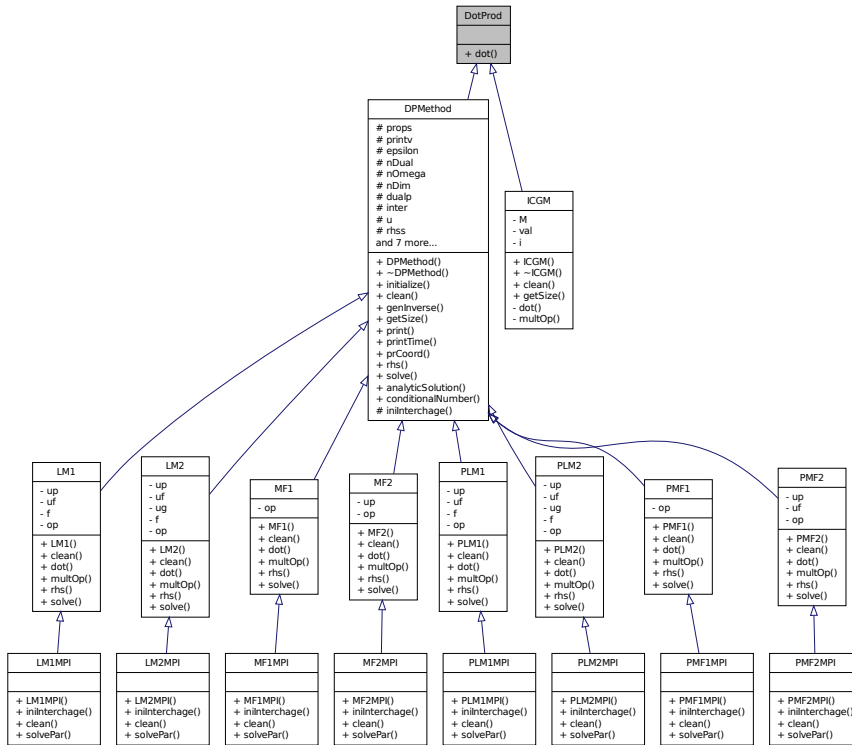
The documentation for this class was generated from the following file:

- [Disc15.hpp](#)

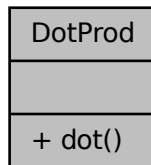
7.12 DotProd Class Reference

```
#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 Idouble DotProd::dot (Idouble * x, Idouble * y) [pure virtual]`

Implemented in [LM2](#), [PLM1](#), [PLM2](#), [LM1](#), [MF2](#), [PMF2](#), [PMF1](#), and [MF1](#).

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

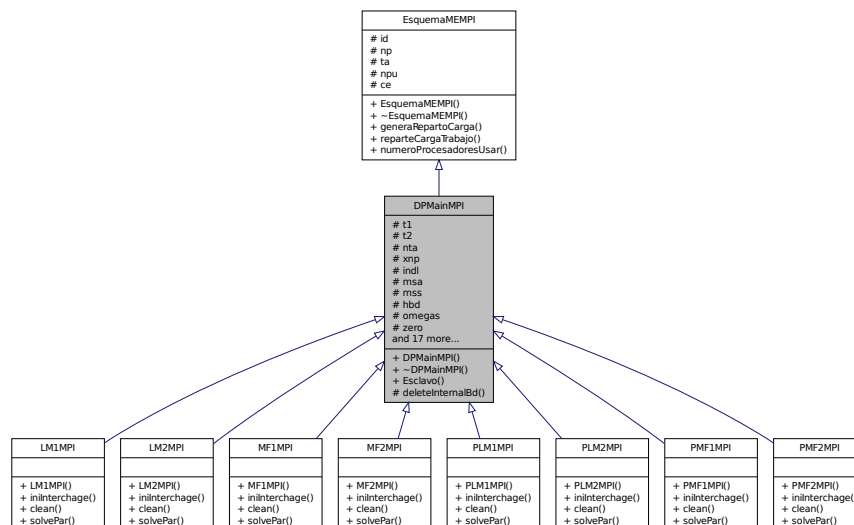
- [DotProd.hpp](#)

7.13 DPMainMPI Class Reference

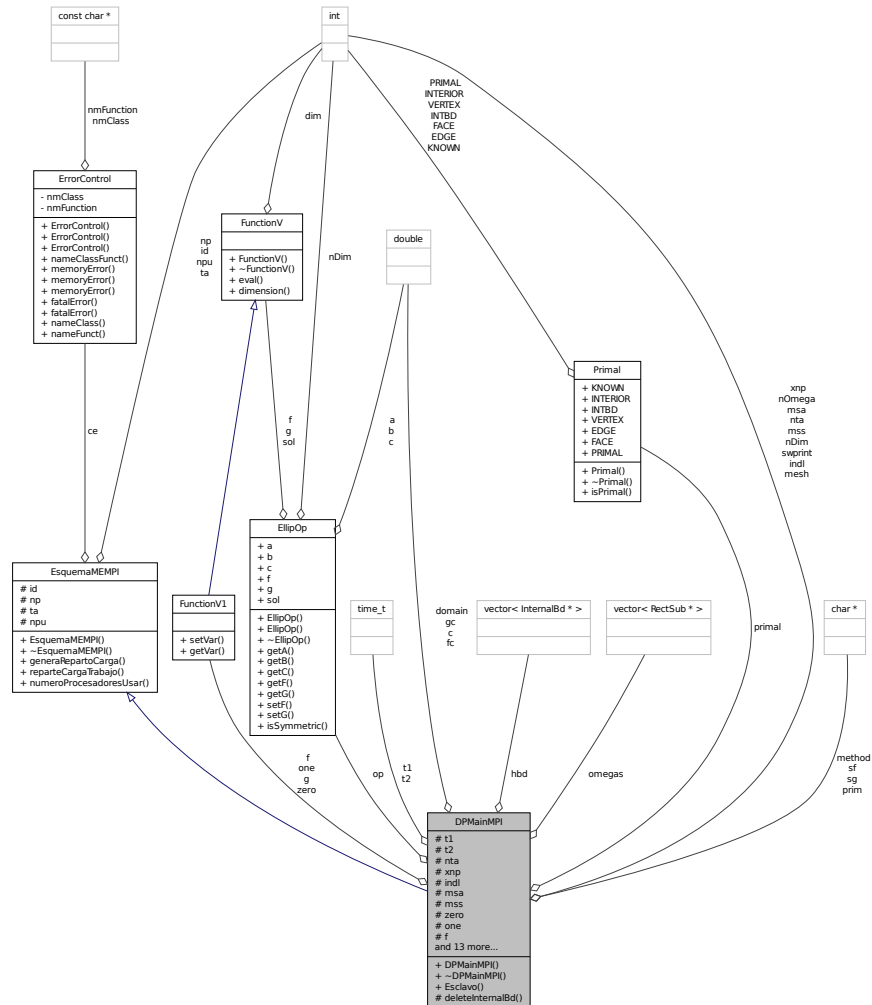
Clase base para definir a los metodos DVS-DDM.

```
#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-Eslavo 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 Idouble DPMainMPI::c [protected]****7.13.4.2 Idouble** DPMainMPI::domain [protected]****7.13.4.3 FunctionV1* DPMainMPI::f [protected]****7.13.4.4 Idouble DPMainMPI::fc [protected]****7.13.4.5 FunctionV1 * DPMainMPI::g [protected]****7.13.4.6 Idouble 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]

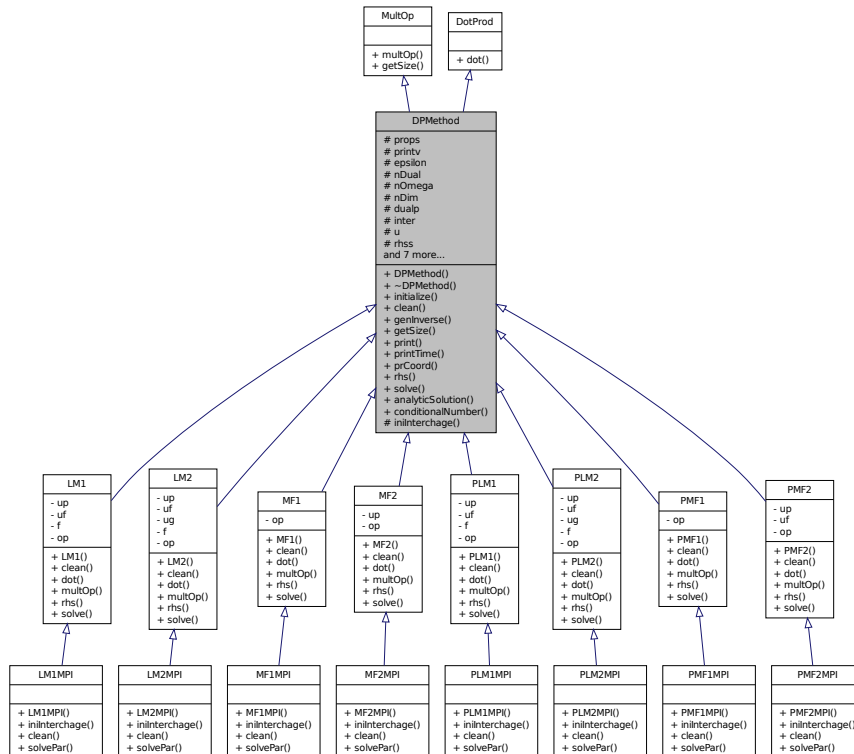
The documentation for this class was generated from the following files:

- [DPMainMPI.hpp](#)
- [DPMainMPI.cpp](#)

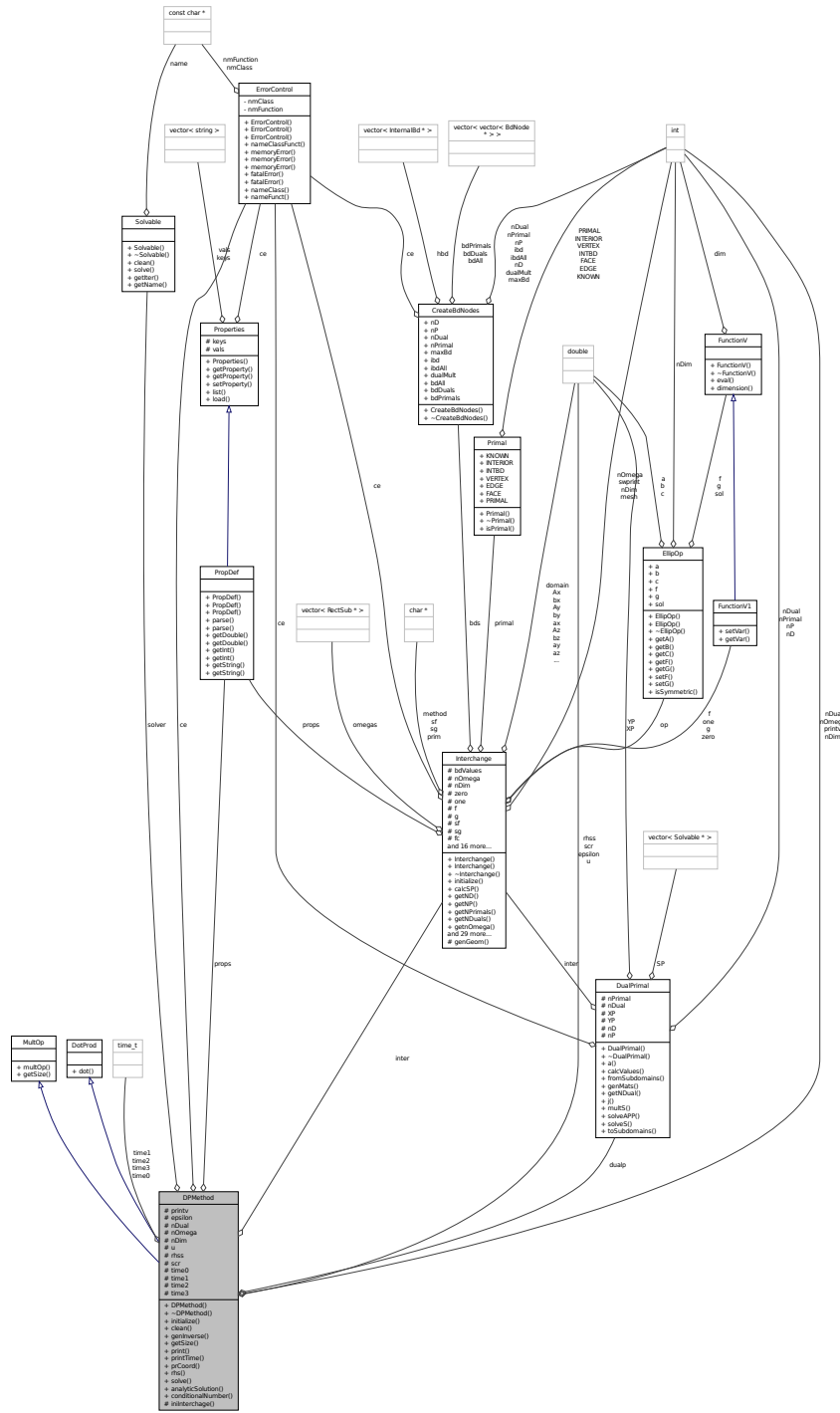
7.14 DPMMethod Class Reference

#include <DPMMethod.hpp>

Inheritance diagram for DPMMethod:



Collaboration diagram for DPMMethod:



Public Member Functions

- [DPMMethod \(PropDef &props\)](#)
- [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 symmetric)
 - Calcula el numero de condicionamiento.*

Protected Member Functions

- virtual void `iniInterchange` (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 `DPMMethod::DPMMethod (PropDef & props)` [inline]

7.14.1.2 `virtual DPMMethod::~~DPMMethod ()` [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 (Idouble * x)`

7.14.2.9 `void DPMethod::print (Idouble * 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*** `DMethod::dualp` [protected]

7.14.3.3 **Idouble** `DMethod::epsilon` [protected]

7.14.3.4 **Interchange*** `DMethod::inter` [protected]

7.14.3.5 **int** `DMethod::nDim` [protected]

7.14.3.6 **int** `DMethod::nDual` [protected]

7.14.3.7 **int** `DMethod::nOmega` [protected]

7.14.3.8 **int** `DMethod::printv` [protected]

7.14.3.9 **PropDef*** `DMethod::props` [protected]

7.14.3.10 **Idouble*** `DMethod::rhss` [protected]

7.14.3.11 **Idouble*** `DMethod::scr` [protected]

7.14.3.12 **Solvable*** `DMethod::solver` [protected]

7.14.3.13 **time_t** `DMethod::time0` [protected]

7.14.3.14 **time_t** `DMethod::time1` [protected]

7.14.3.15 **time_t** `DMethod::time2` [protected]

7.14.3.16 **time_t** `DMethod::time3` [protected]

7.14.3.17 **Idouble*** `DMethod::u` [protected]

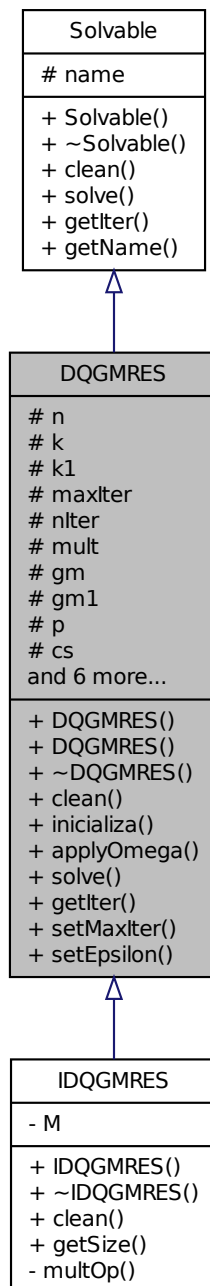
The documentation for this class was generated from the following files:

- [DMethod.hpp](#)
- [DMethod.cpp](#)

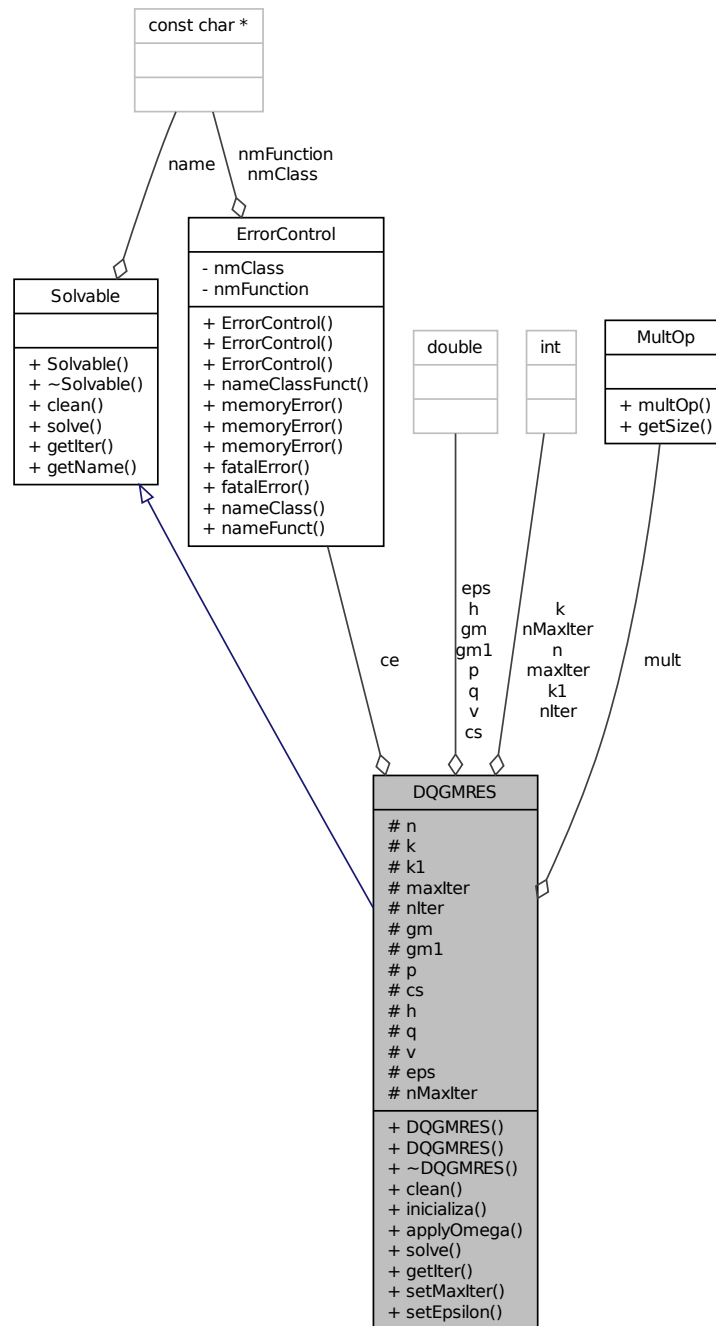
7.15 DQGMRES Class Reference

```
#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 (Idouble ep) [inline]

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

7.15.2.7 void DQGMRES::solve (Idouble * x, Idouble * b) [virtual]

Implements [Solvable](#).

7.15.3 Member Data Documentation

7.15.3.1 ErrorControl DQGMRES::ce [protected]

7.15.3.2 Idouble** DQGMRES::cs [protected]

7.15.3.3 Idouble DQGMRES::eps [protected]

7.15.3.4 Idouble DQGMRES::gm [protected]

7.15.3.5 Idouble DQGMRES::gm1 [protected]

7.15.3.6 Idouble** 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 Idouble** DQGMRES::p [protected]

7.15.3.15 Idouble** DQGMRES::q [protected]

7.15.3.16 Idouble* DQGMRES::v [protected]

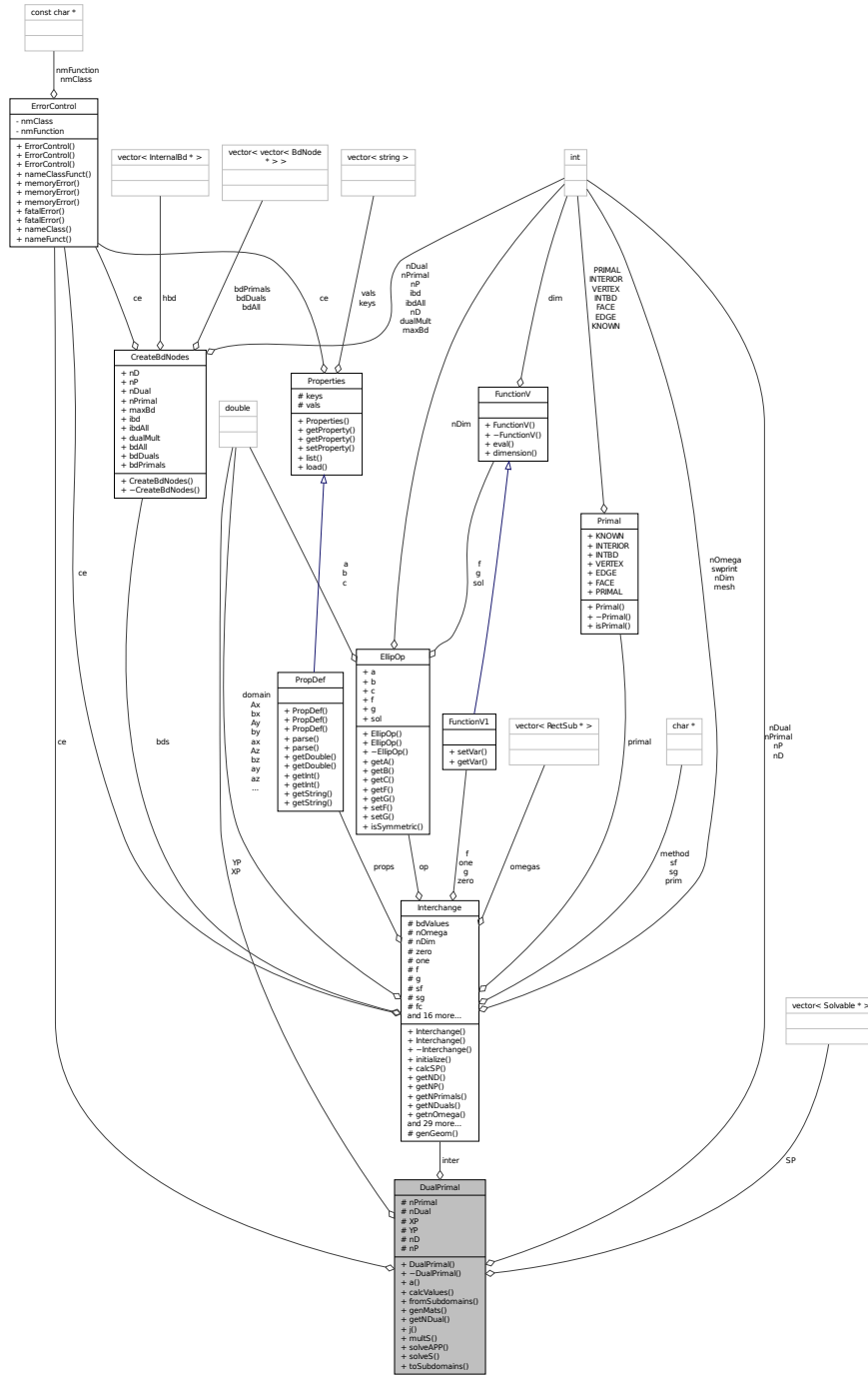
The documentation for this class was generated from the following files:

- [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 (Idouble * *u*, Idouble * *v*)

7.16.2.10 void DualPrimal::toSubdomains (int *sc*, Idouble * *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]

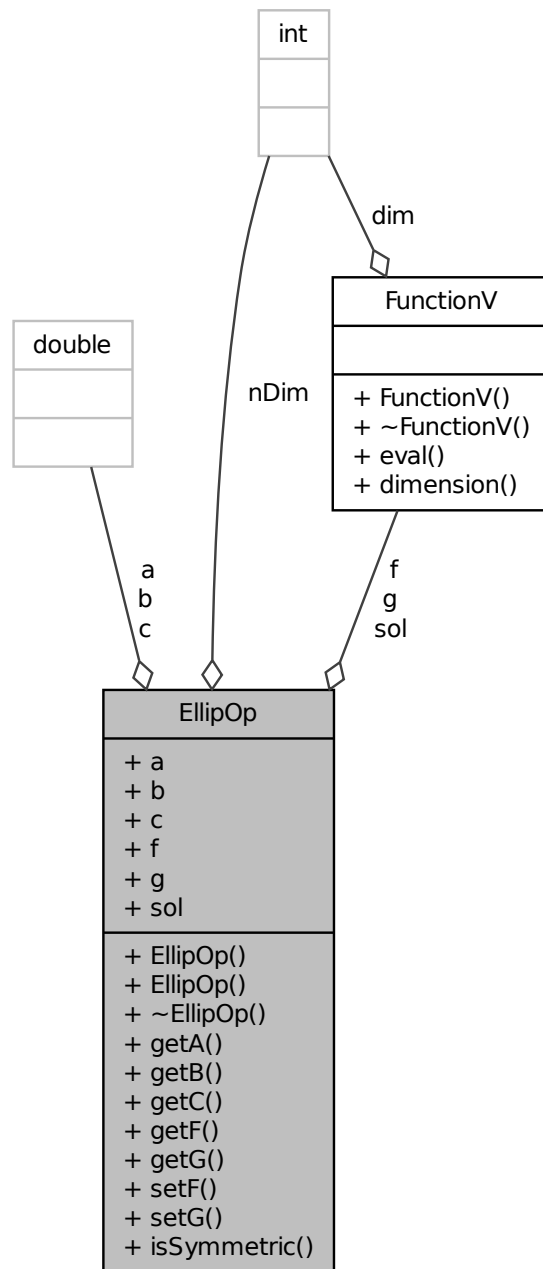
The documentation for this class was generated from the following files:

- [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 `Idouble*` `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:

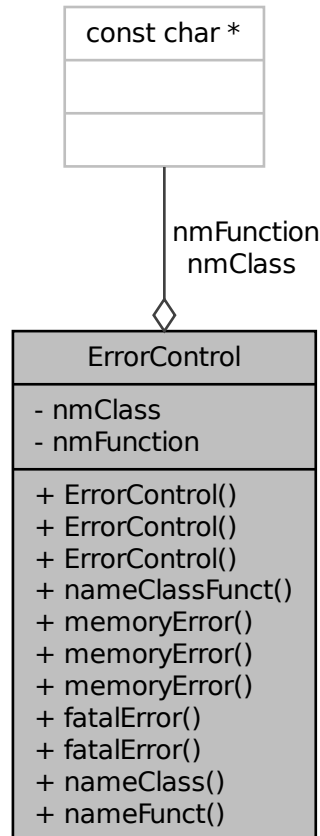
- [EllipOp.hpp](#)

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 [nameClassFuncnt](#) (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 [nameFuncnt](#) (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

<i>clas</i>	Class name
-------------	------------

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

Class Constructor

Parameters

<i>clas</i>	Class name
<i>fun</i>	Function name

7.18.3 Member Function Documentation

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

Fatal error.

Parameters

<i>cod</i>	Error code
------------	------------

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

Fatal error.

Parameters

<i>cod</i>	Error code
<i>txt</i>	Text for user

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

No memory for this request

Parameters

<i>var</i>	Var name
------------	----------

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

No memory for this request

Parameters

<i>var</i>	Var name
<i>i</i>	Index number

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

No memory for this request

Parameters

<i>var</i>	Var name
<i>func</i>	Function name

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

Set name of class

Parameters

<i>clas</i>	Class name
-------------	------------

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

Name of class and function

Parameters

<i>clas</i>	Class name
<i>func</i>	Function name

7.18.3.8 void ErrorControl::nameFunc (const char * *func*)

Set name of function

Parameters

<i>func</i>	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:

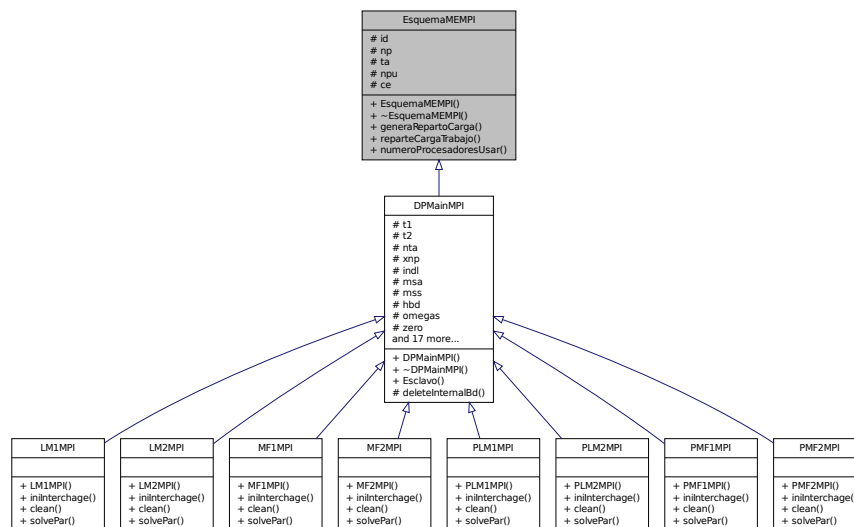
- [ErrorControl.hpp](#)
- [ErrorControl.cpp](#)

7.19 EsquemaMEMPI Class Reference

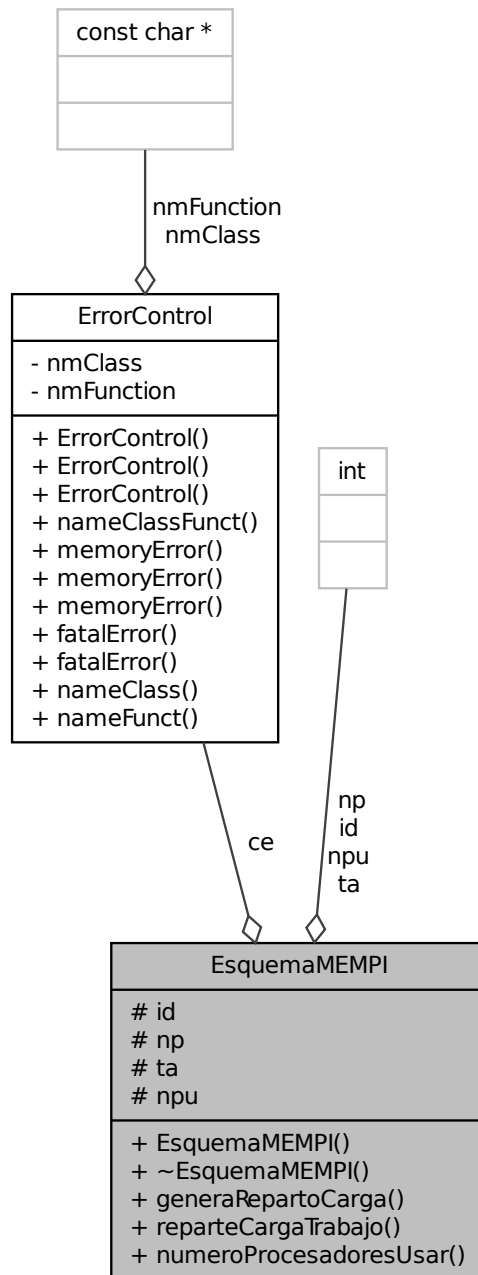
Clase base para definir el Esquema Maestro-Eslavo en MPI.

```
#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-Eslavo en MPI.

Clase base para definir el Esquema Maestro-Eslavo para programar en paralelo mediante el paso de mensajes usando MPI, donde el primer procesador (id = 0) es el nodo maestro y el resto son los nodos esclavos. Las tareas se pueden repartir de manera 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

<i>id</i>	Identificador
<i>np</i>	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

<i>n</i>	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-Eslavo

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

Reparte la carga de trabajo entre los nodos esclavos.

Parameters

<i>np</i>	Numero de procesador esclavo
<i>st</i>	Indice de tarea dentro del nodo esclavo
<i>tarea</i>	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.

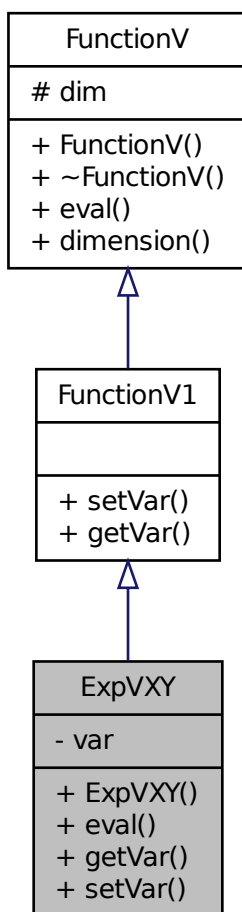
The documentation for this class was generated from the following files:

- [EsquemaMEMPI.hpp](#)
- [EsquemaMEMPI.cpp](#)

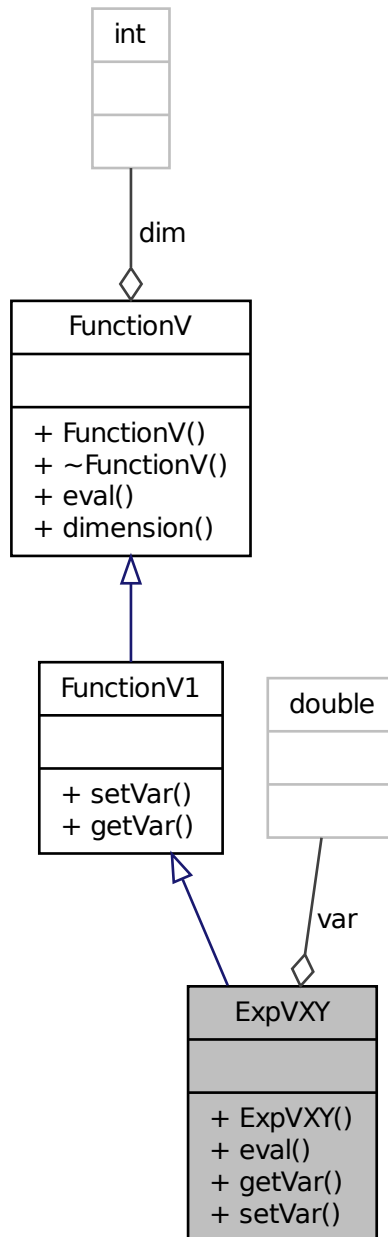
7.20 ExpVXY Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.20.1 Constructor & Destructor Documentation

7.20.1.1 `ExpVXY::ExpVXY (ldouble b)` [`inline`]

7.20.2 Member Function Documentation

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

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

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

Implements [FunctionV1](#).

7.20.3 Member Data Documentation

7.20.3.1 `ldouble ExpVXY::var` [`private`]

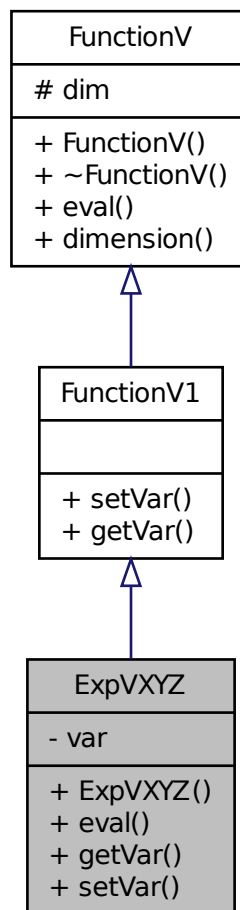
The documentation for this class was generated from the following file:

- [ExpVXY.hpp](#)

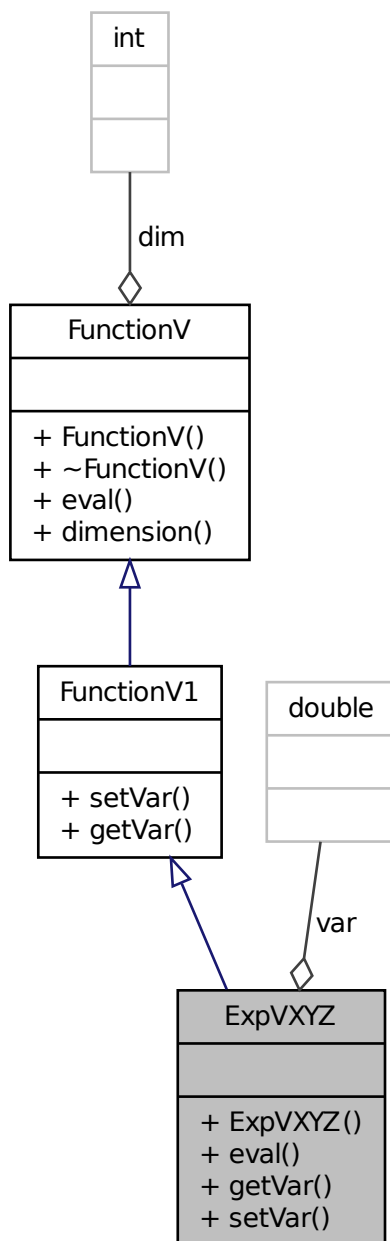
7.21 ExpVXYZ Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.21.1 Constructor & Destructor Documentation

7.21.1.1 `ExpVXYZ::ExpVXYZ(ldouble b)` [[inline](#)]

7.21.2 Member Function Documentation

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

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

7.21.2.3 `void ExpVXYZ::setVar(ldouble b)` [[inline](#)], [[virtual](#)]

Implements [FunctionV1](#).

7.21.3 Member Data Documentation

7.21.3.1 `ldouble ExpVXYZ::var` [[private](#)]

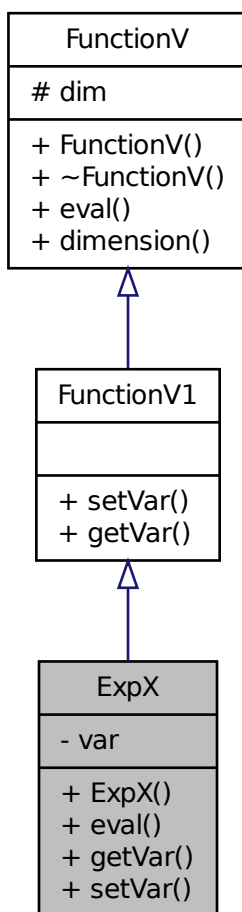
The documentation for this class was generated from the following file:

- [ExpVXYZ.hpp](#)

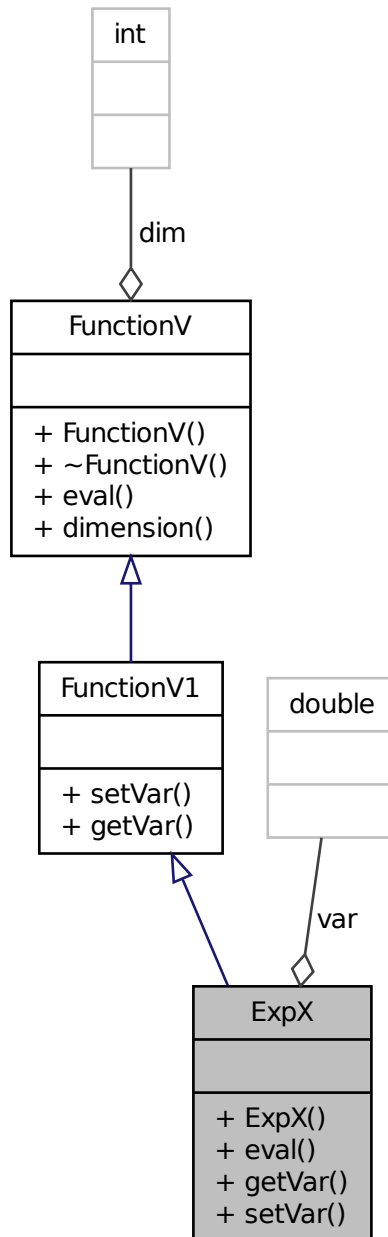
7.22 ExpX Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.22.1 Constructor & Destructor Documentation

7.22.1.1 `ExpX::ExpX (ldouble b)` [[inline](#)]

7.22.2 Member Function Documentation

7.22.2.1 `ldouble ExpX::eval (int d, ldouble * x)` [[inline](#)],[[virtual](#)]

Implements [FunctionV](#).

7.22.2.2 `ldouble ExpX::getVar (void)` [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.22.2.3 `void ExpX::setVar (ldouble b)` [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.22.3 Member Data Documentation

7.22.3.1 `ldouble ExpX::var` [[private](#)]

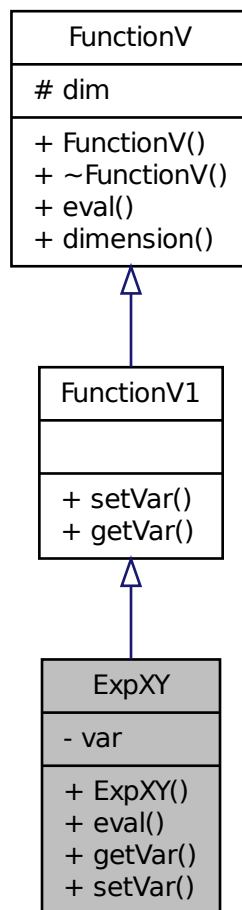
The documentation for this class was generated from the following file:

- [ExpX.hpp](#)

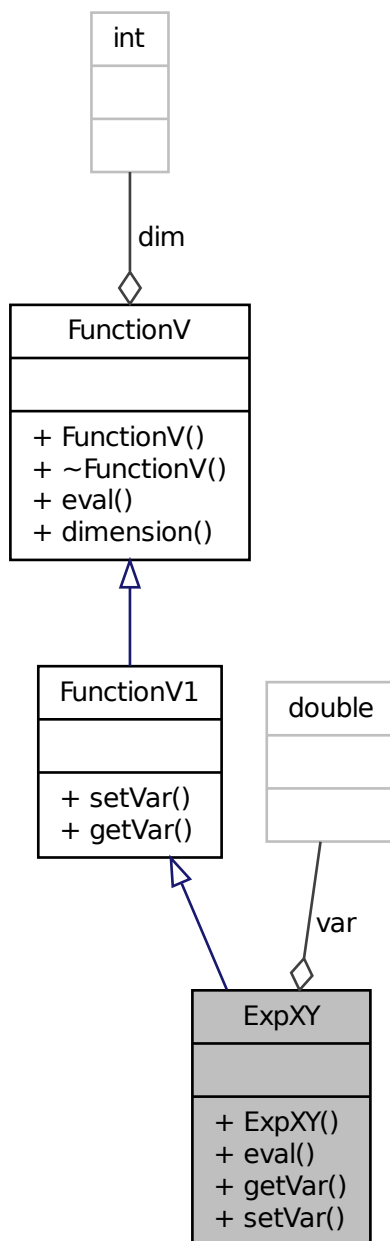
7.23 ExpXY Class Reference

```
#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](#)]

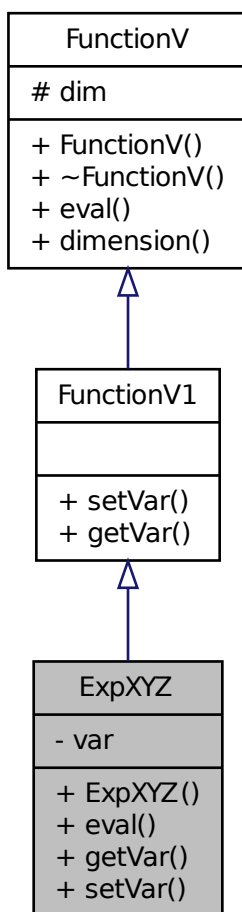
The documentation for this class was generated from the following file:

- [ExpXY.hpp](#)

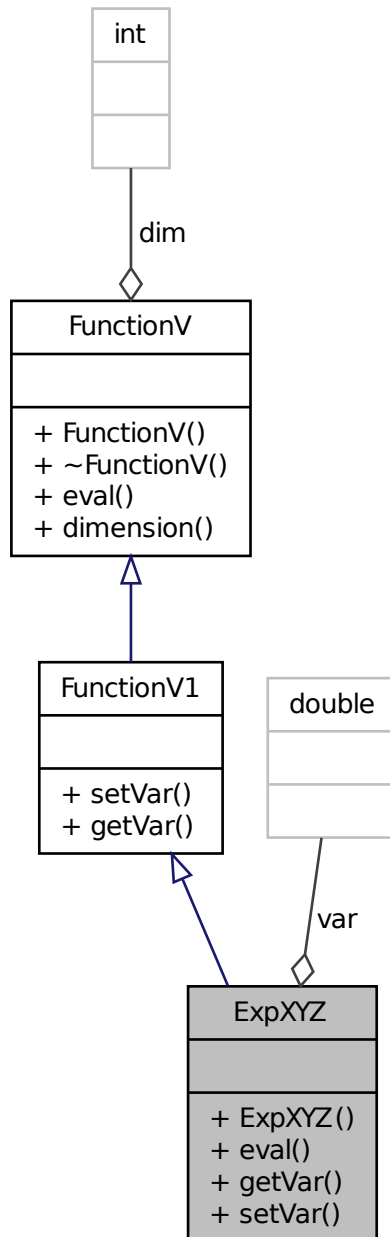
7.24 ExpXYZ Class Reference

```
#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](#)]

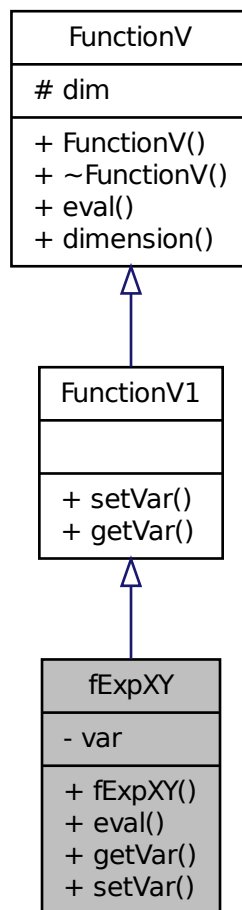
The documentation for this class was generated from the following file:

- [ExpXYZ.hpp](#)

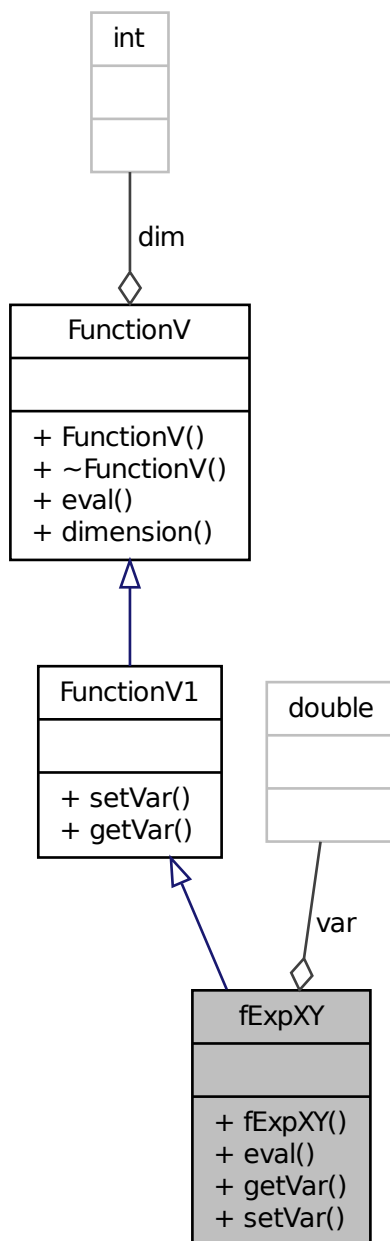
7.25 fExpXY Class Reference

```
#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]`

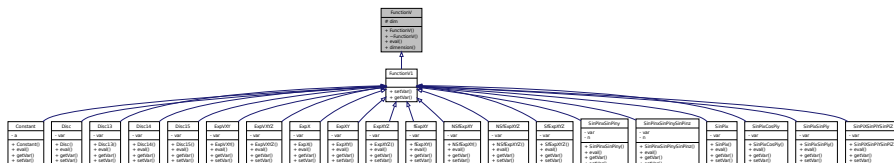
The documentation for this class was generated from the following file:

- [fExpXY.hpp](#)

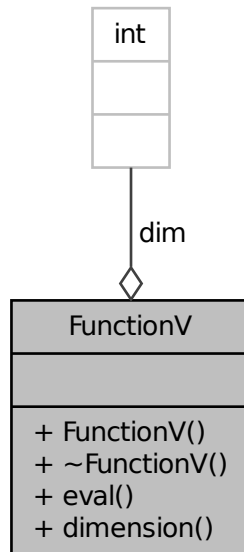
7.26 FunctionV Class Reference

```
#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]

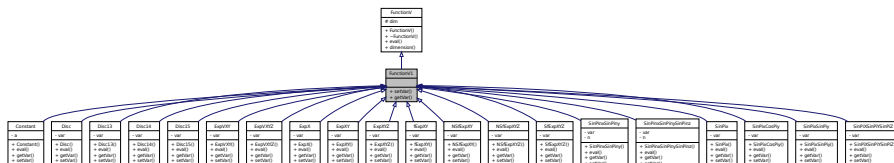
The documentation for this class was generated from the following file:

- [FunctionV.hpp](#)

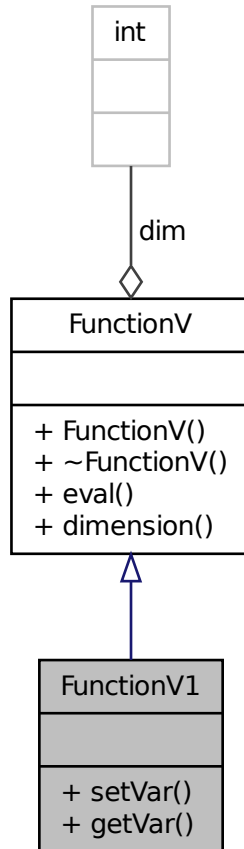
7.27 FunctionV1 Class Reference

```
#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](#).

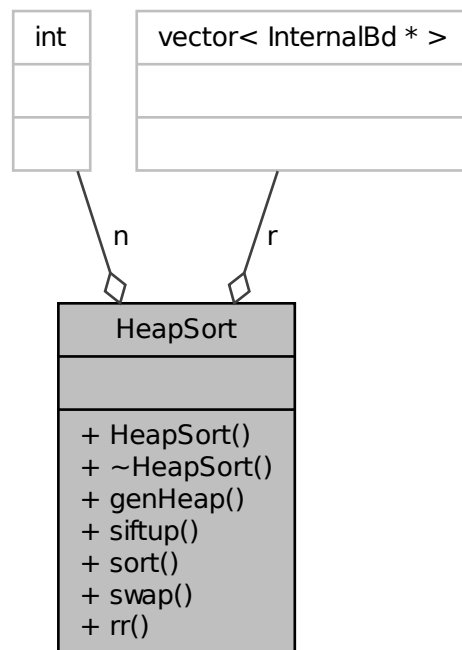
The documentation for this class was generated from the following file:

- [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:

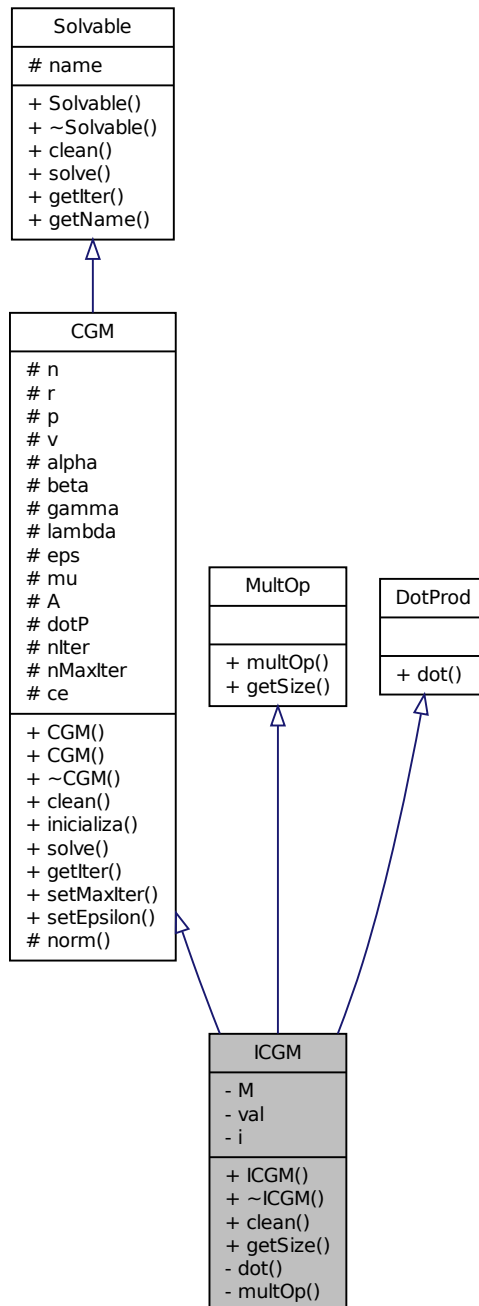
- [HeapSort.hpp](#)

7.29 ICGM Class Reference

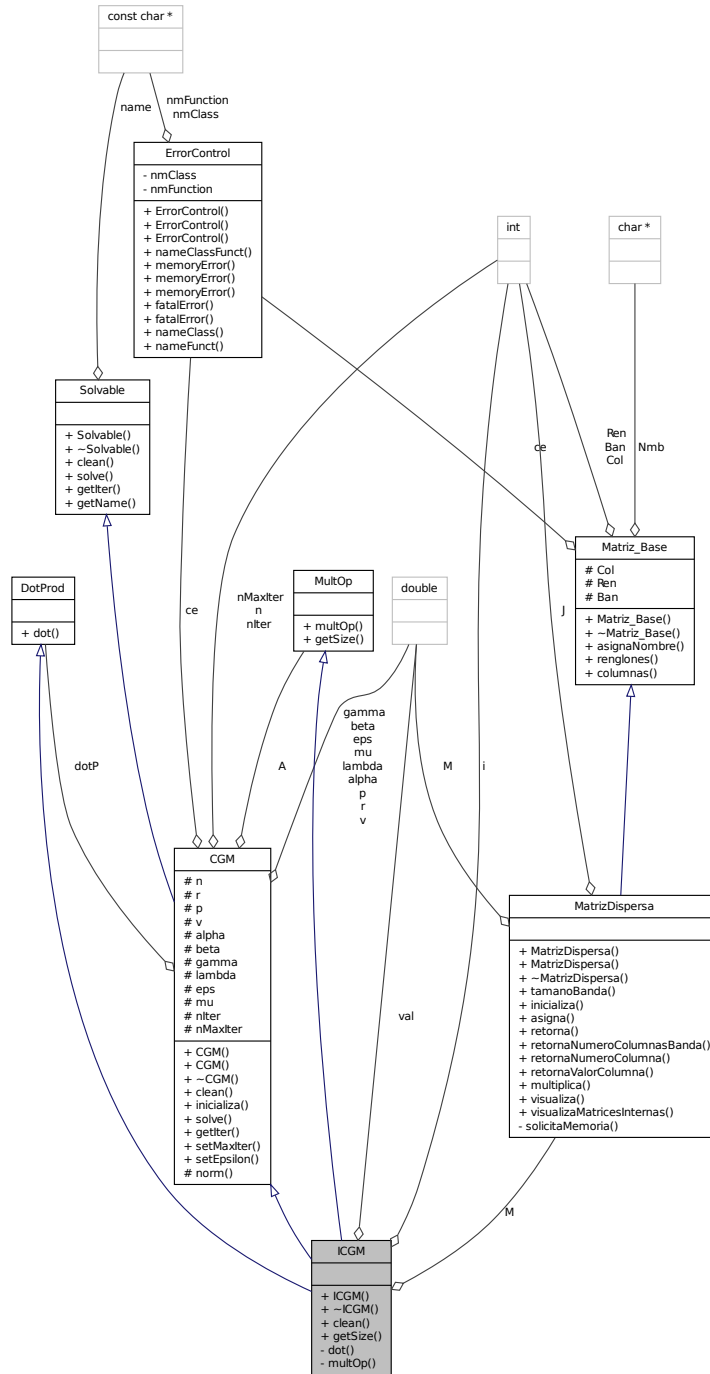
Clase para implementar [CGM](#) con matrices bandadas o dispersas.

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

Inheritance diagram for ICGM:



Collaboration diagram for ICGM:



Public Member Functions

- **ICGM** (int `n`, **MatrizDispersa** *`M`, **Idouble** `eps`, int `iter`)

Constructor 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`)

Multiplca $Au=v$.

Private Attributes

- `MatrizDispersa * M`

Multiplca $Au=v$.

- `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]`

Constructor 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]`

Producto punto.

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]`

Multiplca Au=v.

7.29.4 Member Data Documentation

7.29.4.1 `int ICGM::i [private]`

7.29.4.2 `MatrizDispersa* ICGM::M [private]`

Multiplca Au=v.

7.29.4.3 `ldouble ICGM::val [private]`

Variables temporales.

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

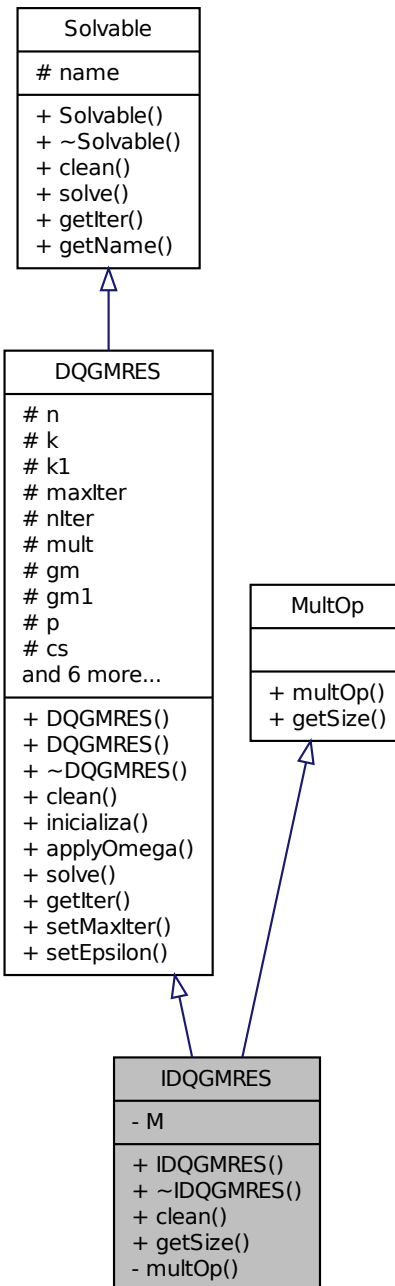
- [ICGM.hpp](#)

7.30 IDQGMRES Class Reference

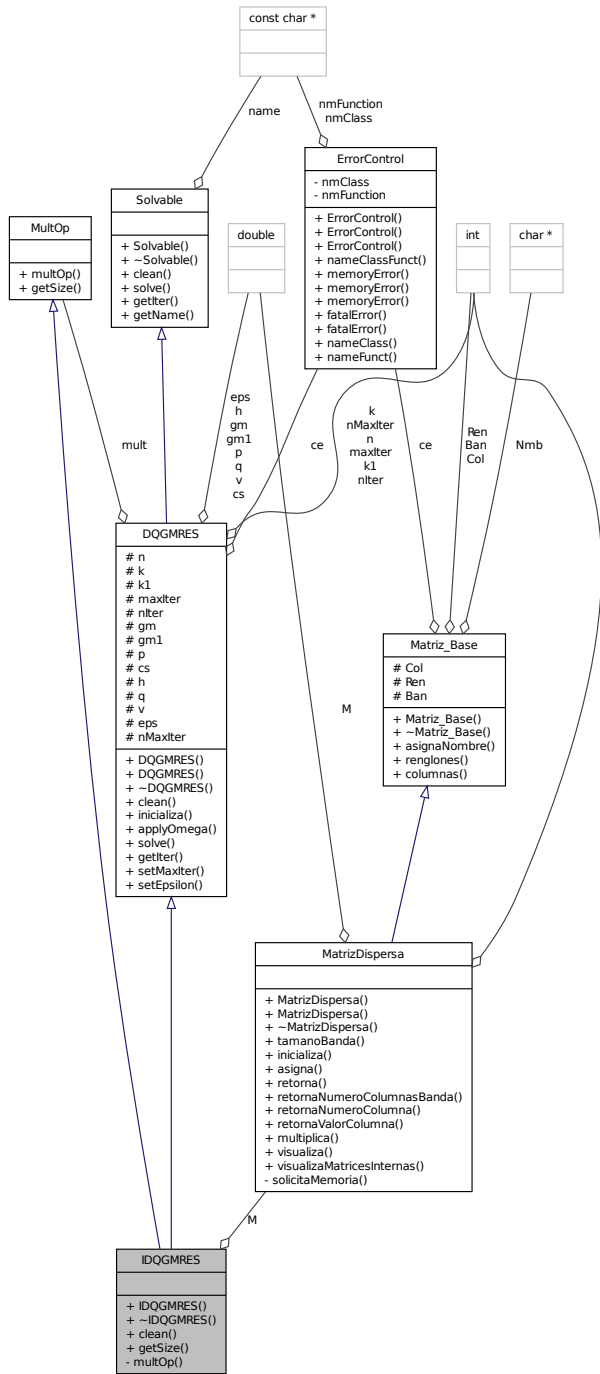
Clase para implementar [DQGMRES](#) con matrices bandadas o dispersas.

```
#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`)

Multiplifica $Au=v$.

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]`

Multiplica $Au=v$.

7.30.4 Member Data Documentation

7.30.4.1 `MatrizDispersa* IDQGMRES::M [private]`

Matriz Bandada o Dispersa.

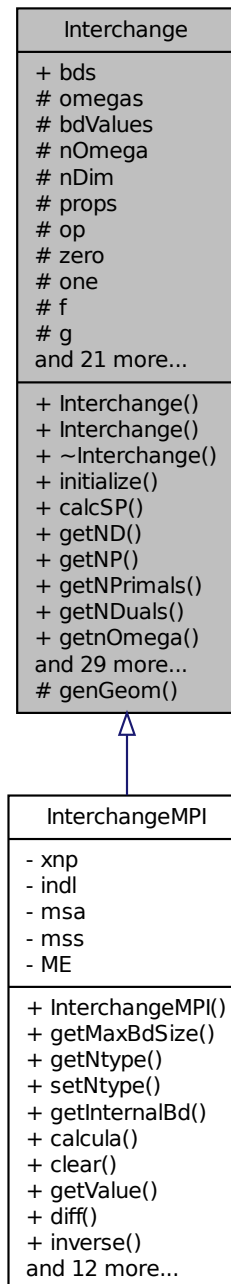
The documentation for this class was generated from the following file:

- [IDQGMRES.hpp](#)

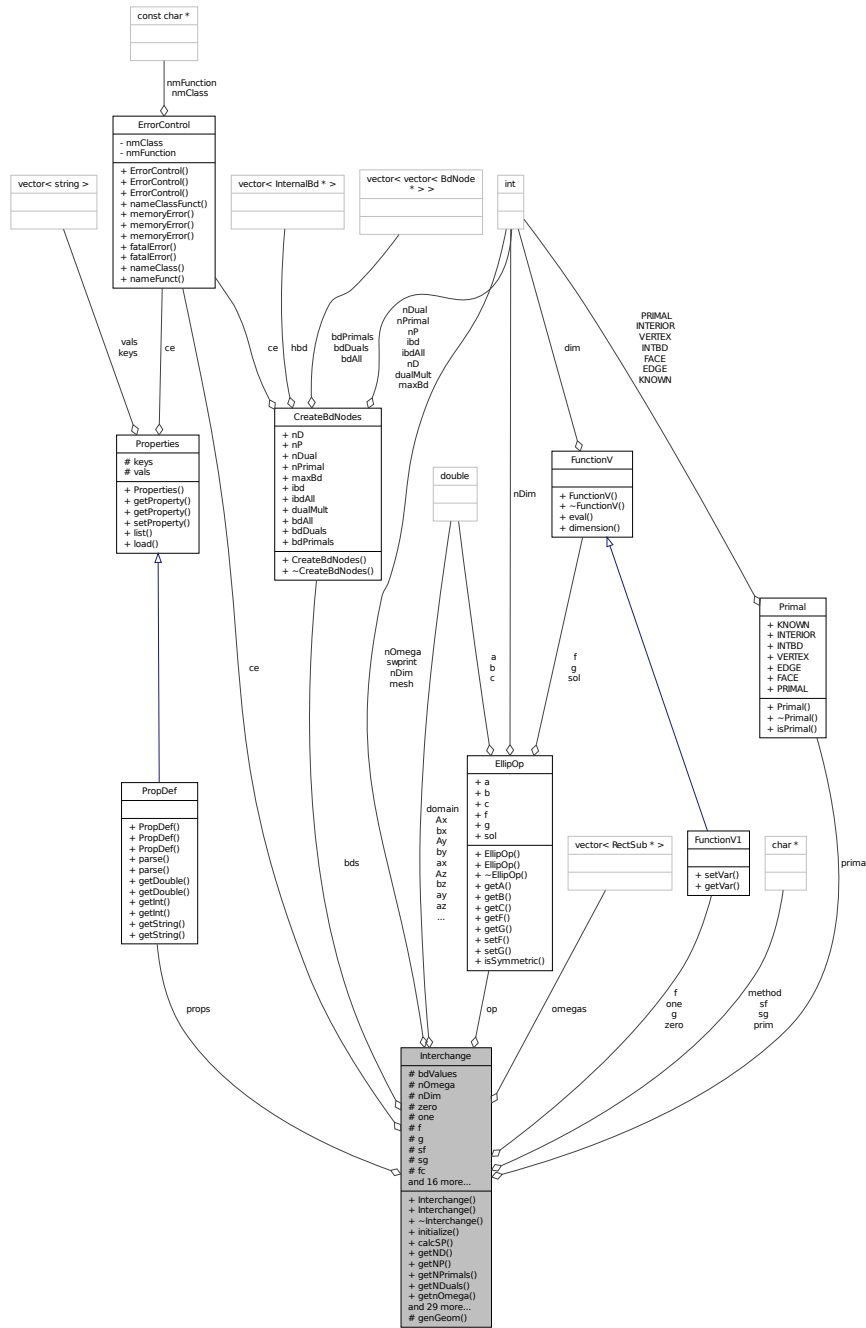
7.31 Interchange Class Reference

```
#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, Idouble 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]

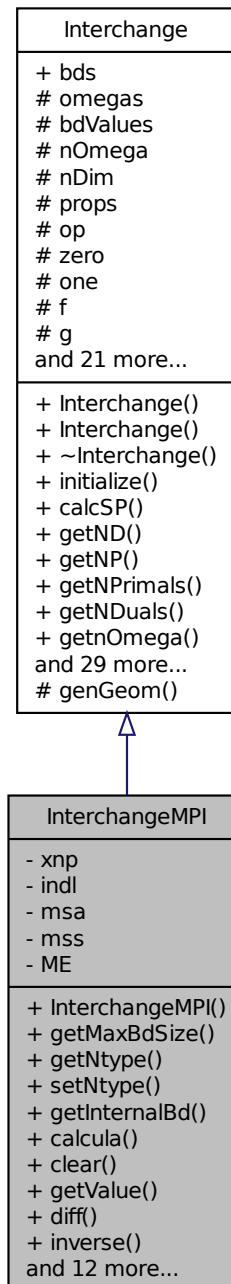
The documentation for this class was generated from the following files:

- [Interchange.hpp](#)
- [Interchange.cpp](#)

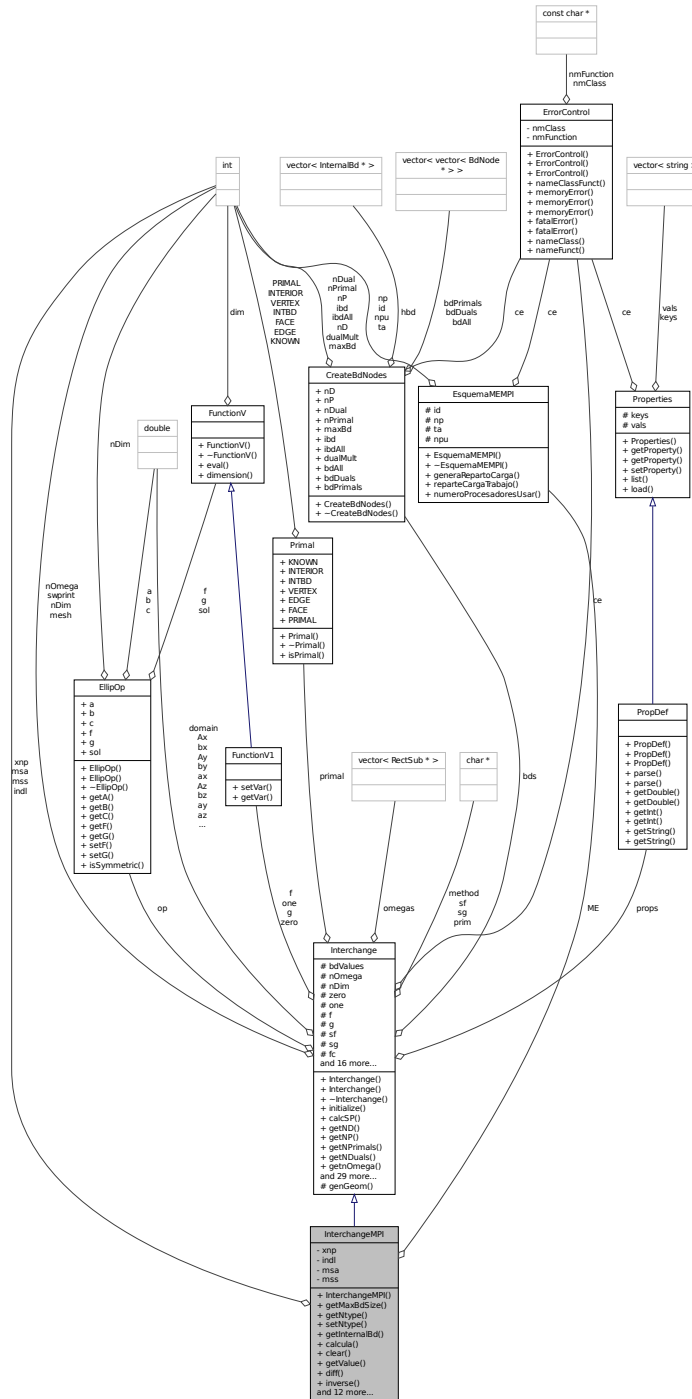
7.32 InterchangeMPI Class Reference

```
#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-Eslavo.

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 `Idouble 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-Eslavo.

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.

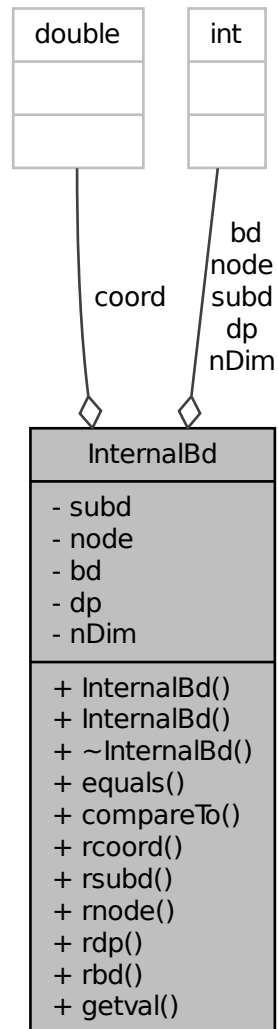
The documentation for this class was generated from the following files:

- [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]`

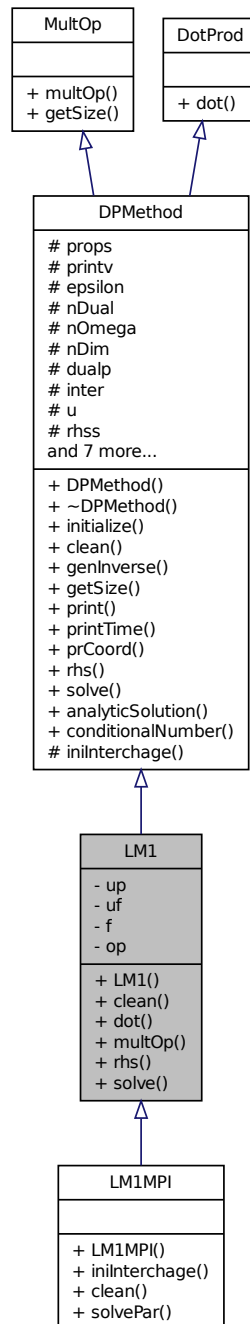
The documentation for this class was generated from the following file:

- [InternalBd.hpp](#)

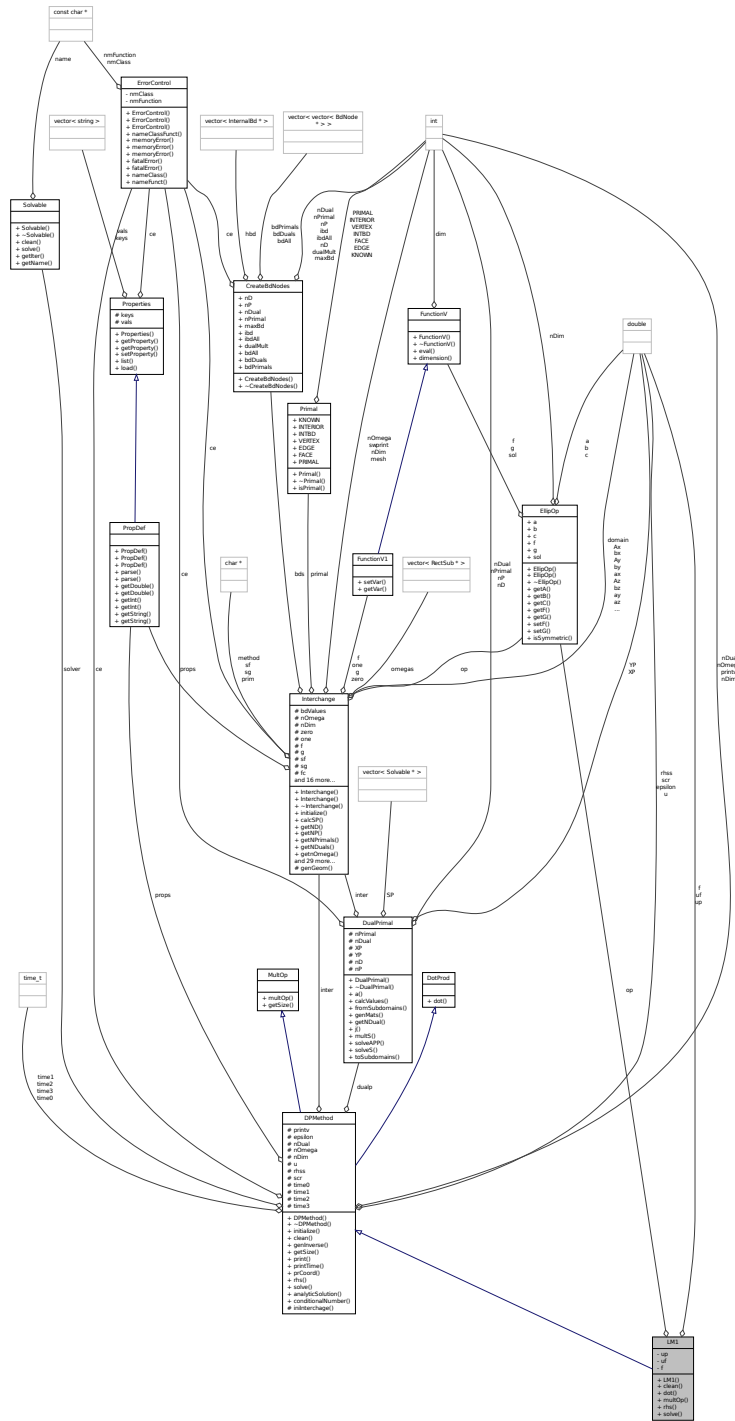
7.34 LM1 Class Reference

```
#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 `Idouble* LM1::f` [private]

7.34.3.2 `EllipOp* LM1::op` [private]

7.34.3.3 `Idouble* LM1::uf` [private]

7.34.3.4 `Idouble* LM1::up` [private]

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

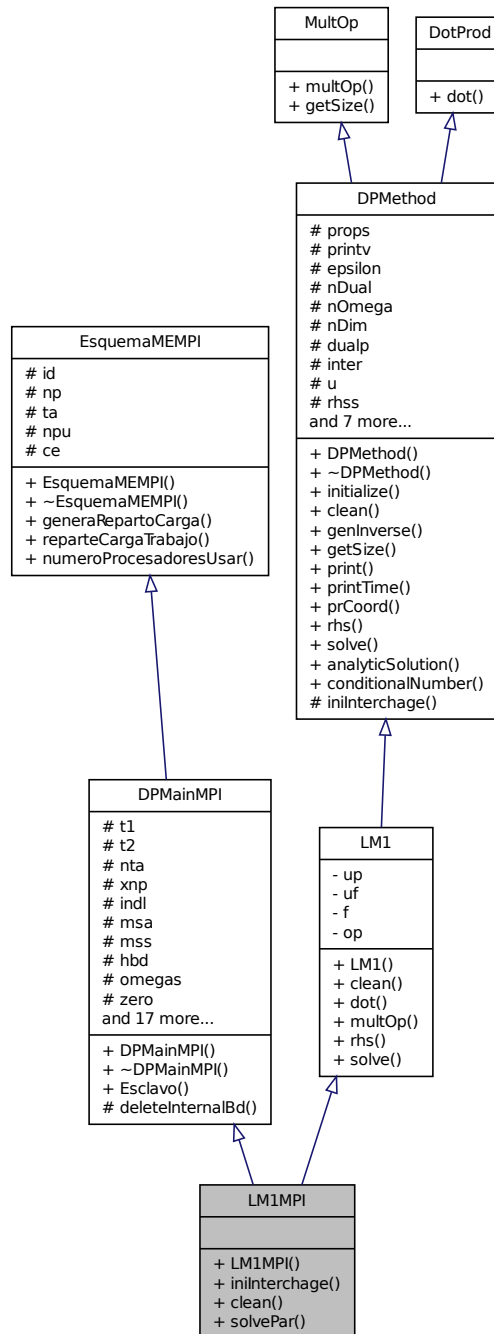
- [LM1.hpp](#)
- [LM1.cpp](#)

7.35 LM1MPI Class Reference

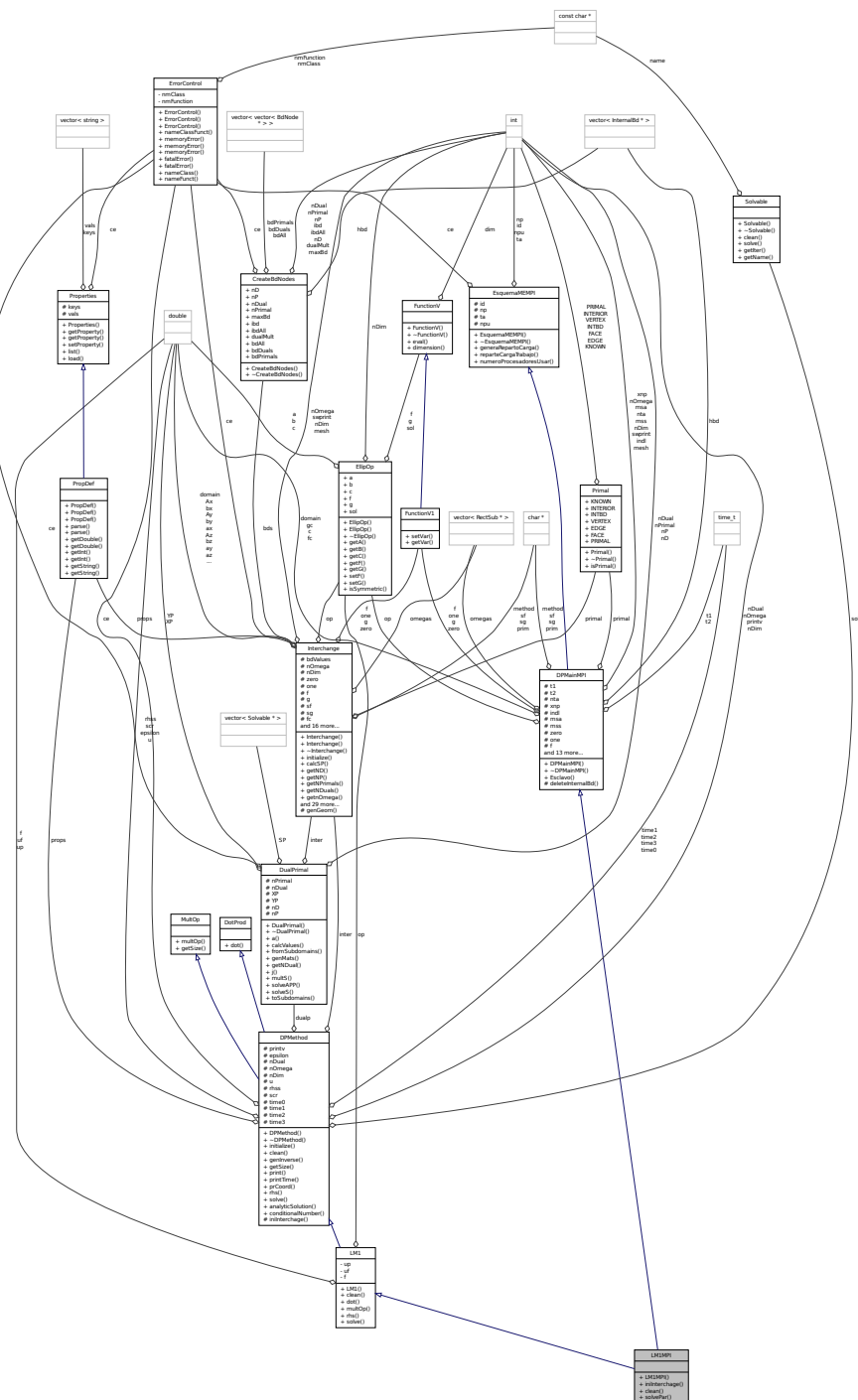
Clase para definir el metodo LM-1 de DVS-DDM.

```
#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.

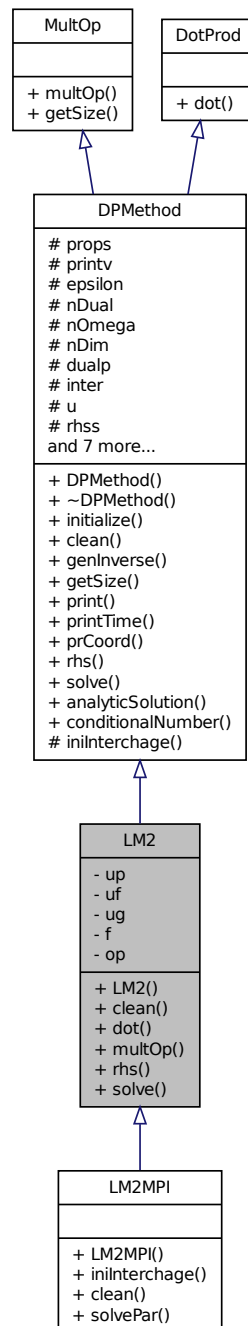
The documentation for this class was generated from the following file:

- [LM1MPI.hpp](#)

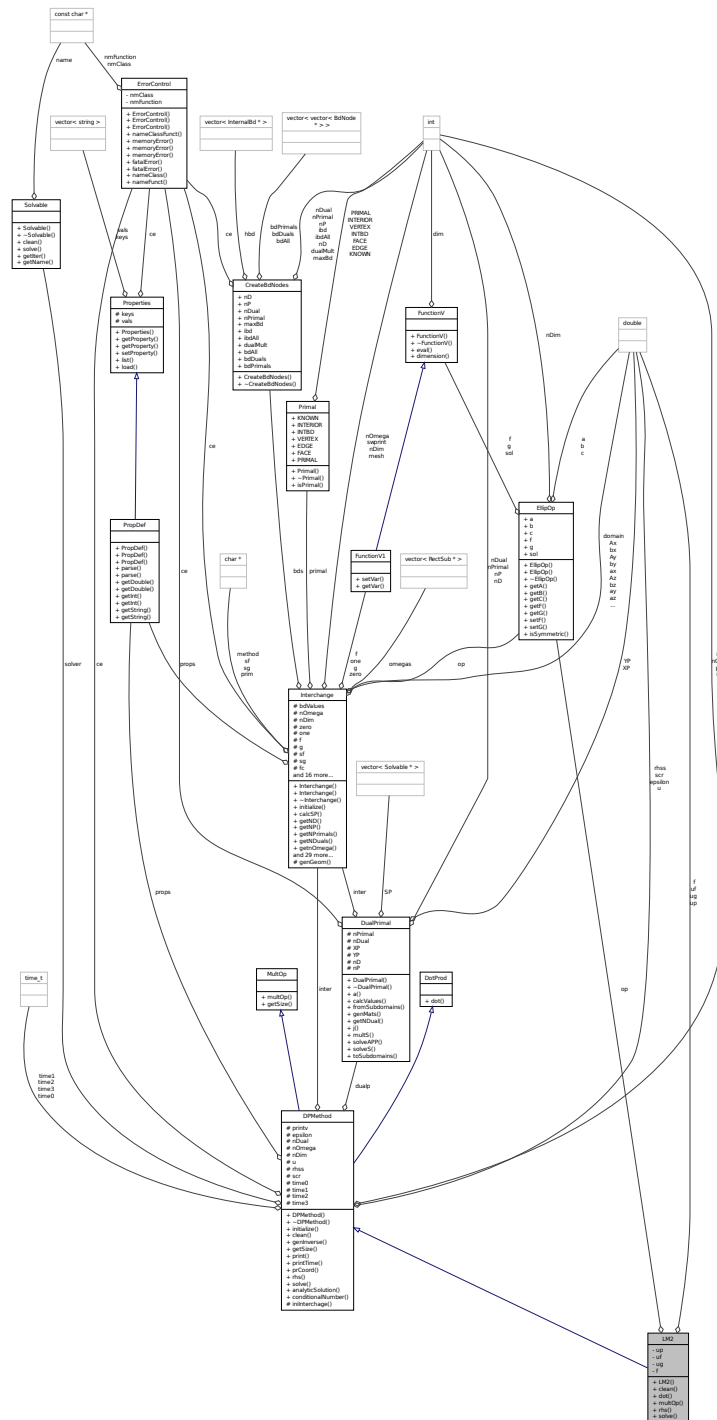
7.36 LM2 Class Reference

```
#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 `Idouble* LM2::f` [private]

7.36.3.2 `EllipOp* LM2::op` [private]

7.36.3.3 `Idouble* LM2::uf` [private]

7.36.3.4 `Idouble* LM2::ug` [private]

7.36.3.5 `Idouble* LM2::up` [private]

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

- [LM2.hpp](#)

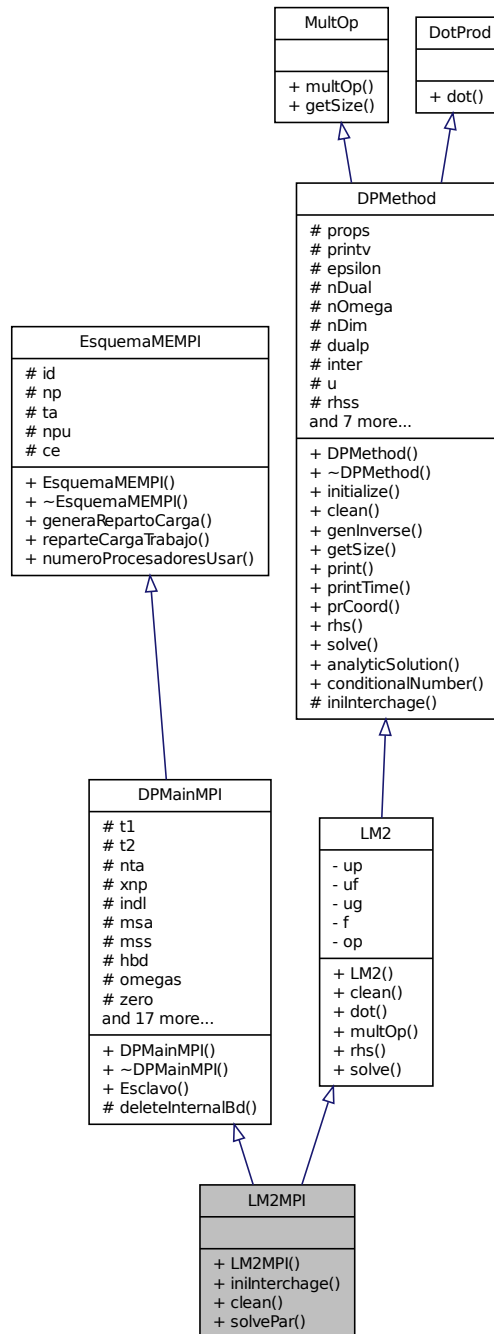
- [LM2.cpp](#)

7.37 LM2MPI Class Reference

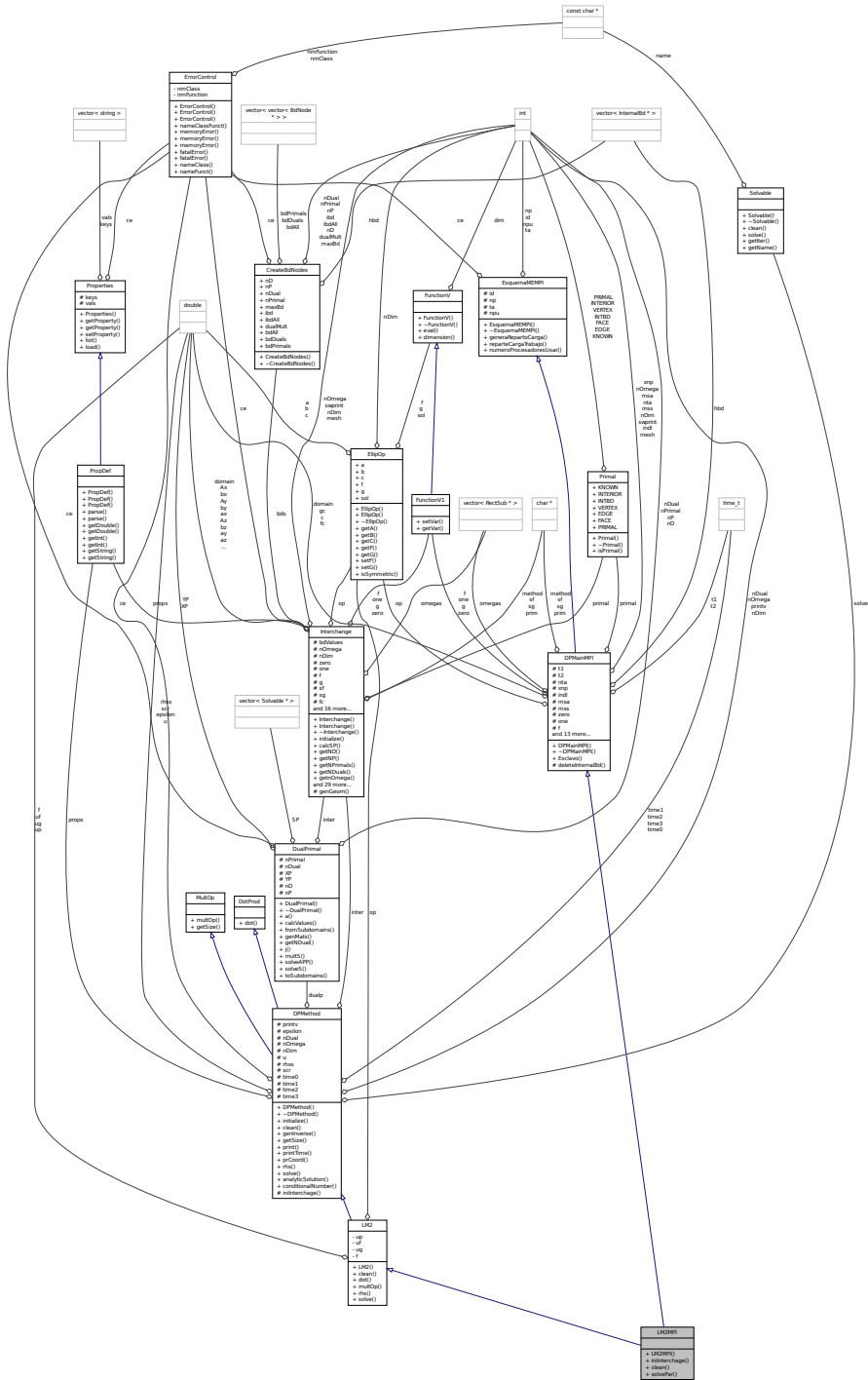
Clase para definir el metodo LM-2 de DVS-DDM.

```
#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.

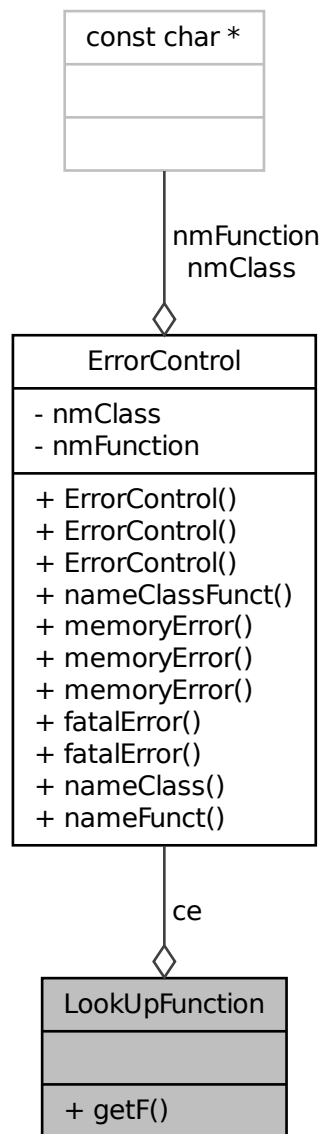
The documentation for this class was generated from the following file:

- [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:

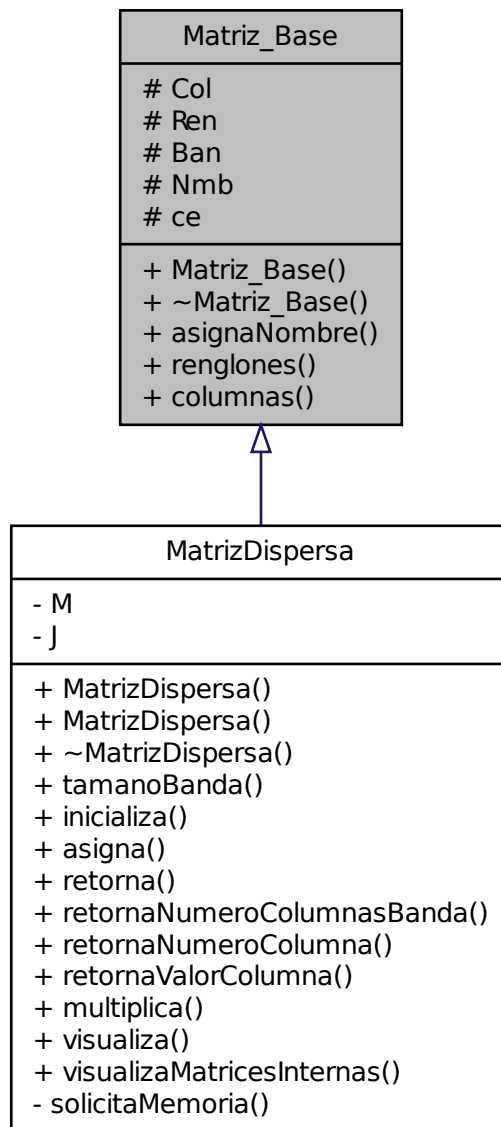
- [LookUpFunction.hpp](#)
- [LookUpFunction.cpp](#)

7.39 Matriz_Base Class Reference

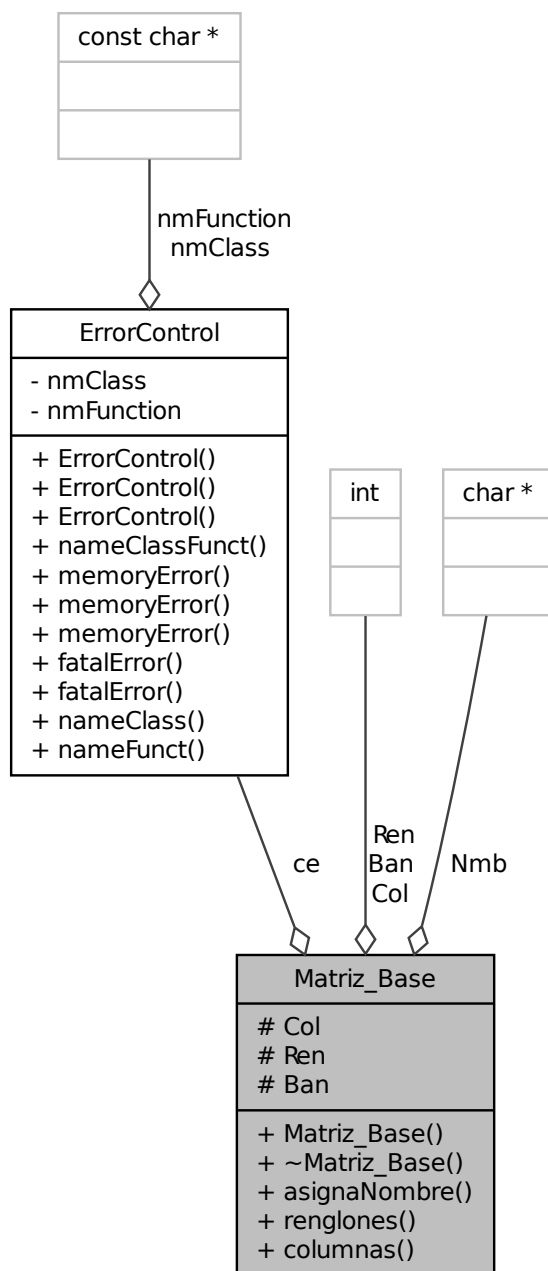
Clase base para el trabajar con matrices.

```
#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

<i>nmb</i>	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.

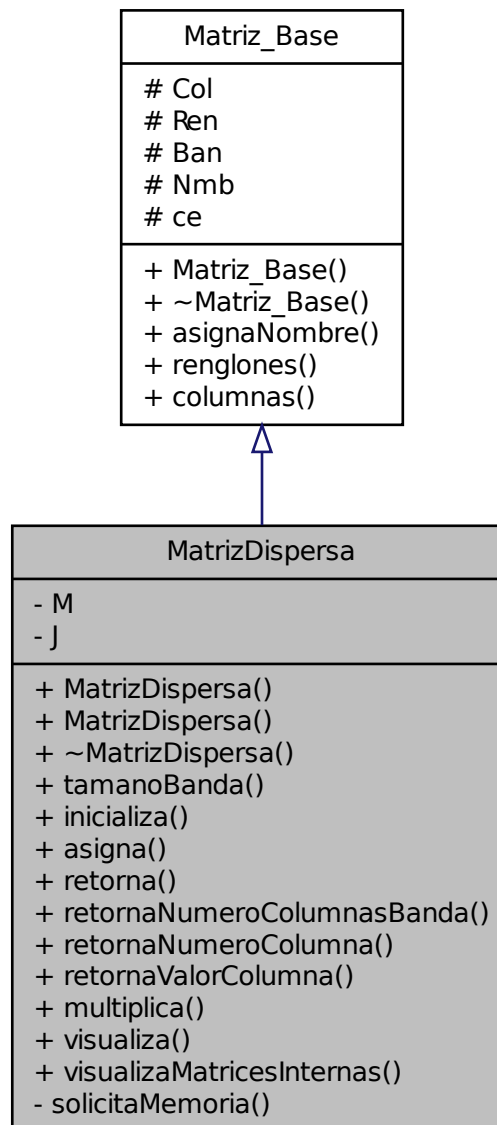
The documentation for this class was generated from the following file:

- [Matriz_Base.hpp](#)

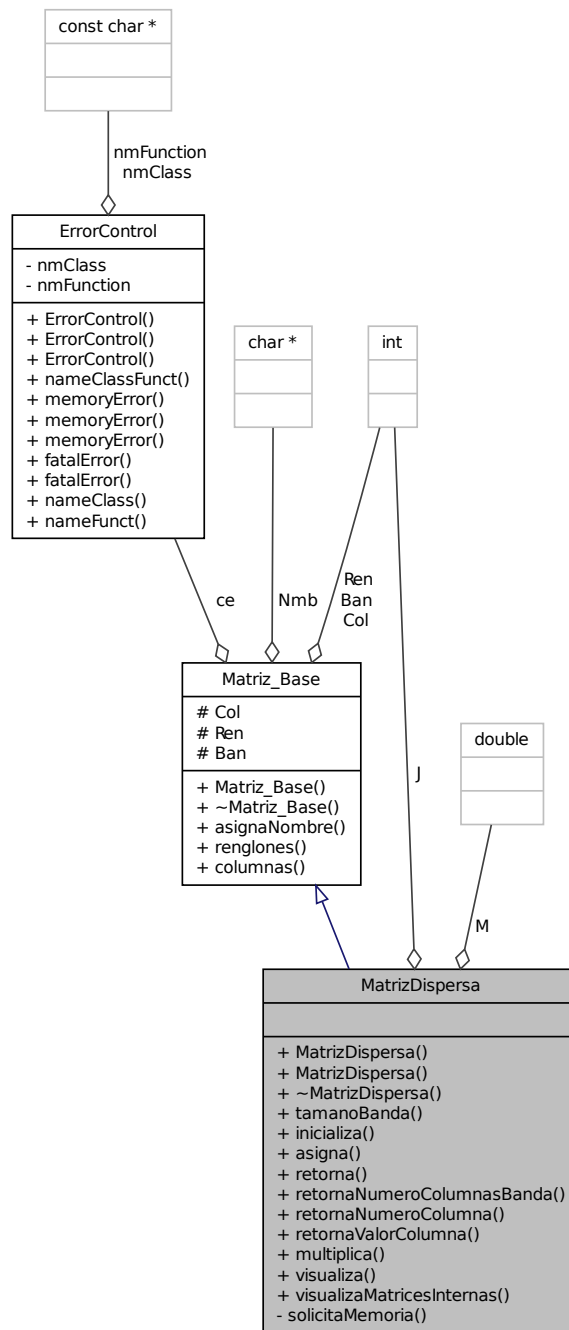
7.40 MatrizDispersa Class Reference

```
#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 tamaño de banda
Multiplicación 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

<i>ren</i>	Numero de renglones de la matriz
<i>col</i>	Numero de columnas de la matriz
<i>ban</i>	Tamaño 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

<i>ren</i>	Numero de renglones de la matriz
<i>col</i>	Numero de columnas de la matriz
<i>ban</i>	Tamaño de la banda
<i>nmb</i>	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

<i>ren</i>	Renglon
<i>col</i>	Columna
<i>val</i>	Valor

7.40.3.2 void MatrizDispersa::inicializa (Idouble val) [inline]

Inicializa la matriz al valor indicado.

Parameters

<i>val</i>	Valor por omision para inicializar la matriz
------------	--

7.40.3.3 void MatrizDispersa::multiplica (Idouble * b, Idouble * r)

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

Parameters

<i>b</i>	Puntero a un Vector
<i>r</i>	Puntero a un Vector

7.40.3.4 Idouble 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

<i>ren</i>	Numero de renglon
<i>col</i>	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

<i>ren</i>	Numero de renglon
<i>ind</i>	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

<i>ren</i>	Numero de renglon
------------	-------------------

Returns

Numero de columnas de la banda para el renglon solicitado

7.40.3.7 `Idouble 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

<i>ren</i>	Numero de renglon
<i>ind</i>	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

<i>ren</i>	Numero de renglones de la matriz
<i>col</i>	Numero de columnas de la matriz
<i>ban</i>	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

<i>tp</i>	(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 `Idouble** MatrizDispersa::M` [private]

Puntero a la matriz de datos.

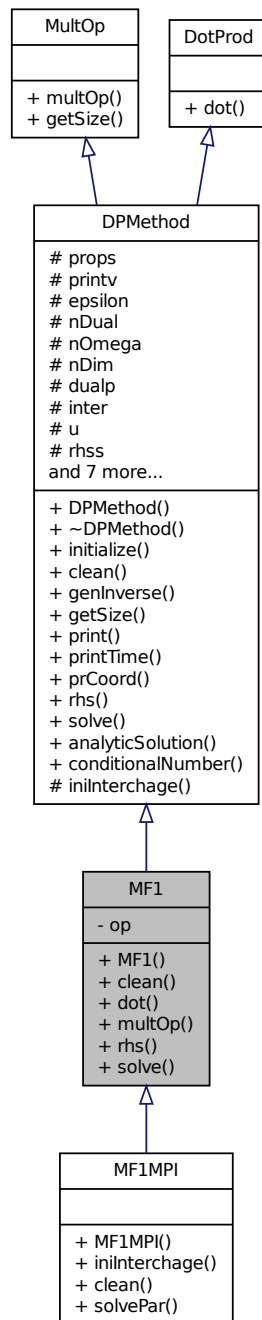
The documentation for this class was generated from the following files:

- [MatrizDispersa.hpp](#)
- [MatrizDispersa.cpp](#)

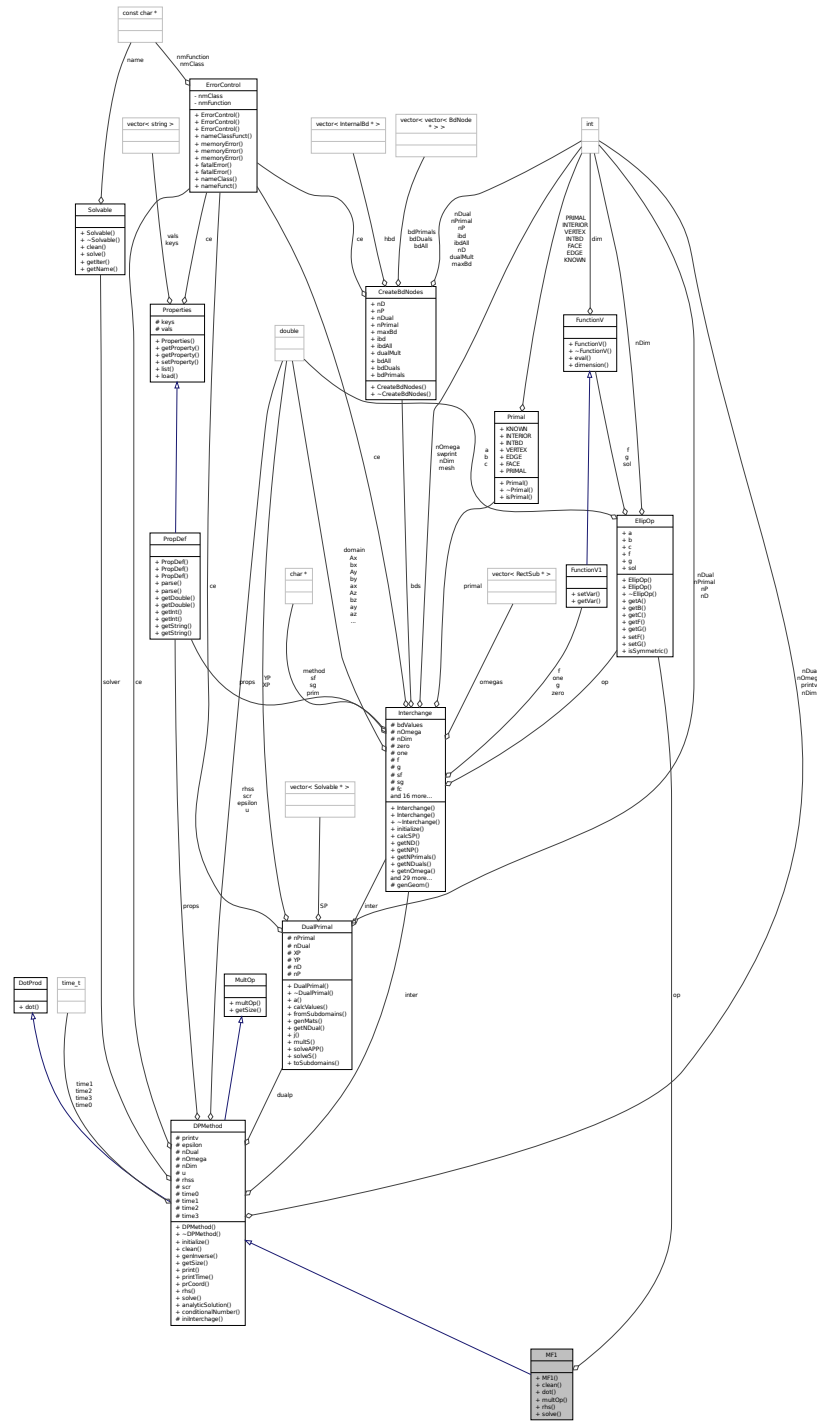
7.41 MF1 Class Reference

```
#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](#)

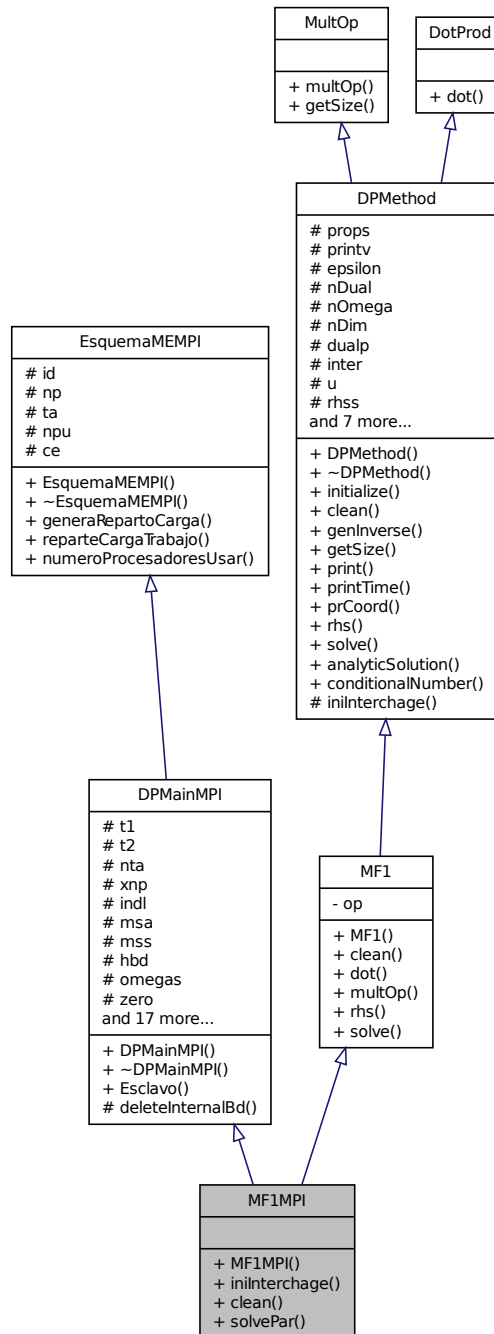
- [MF1.cpp](#)

7.42 MF1MPI Class Reference

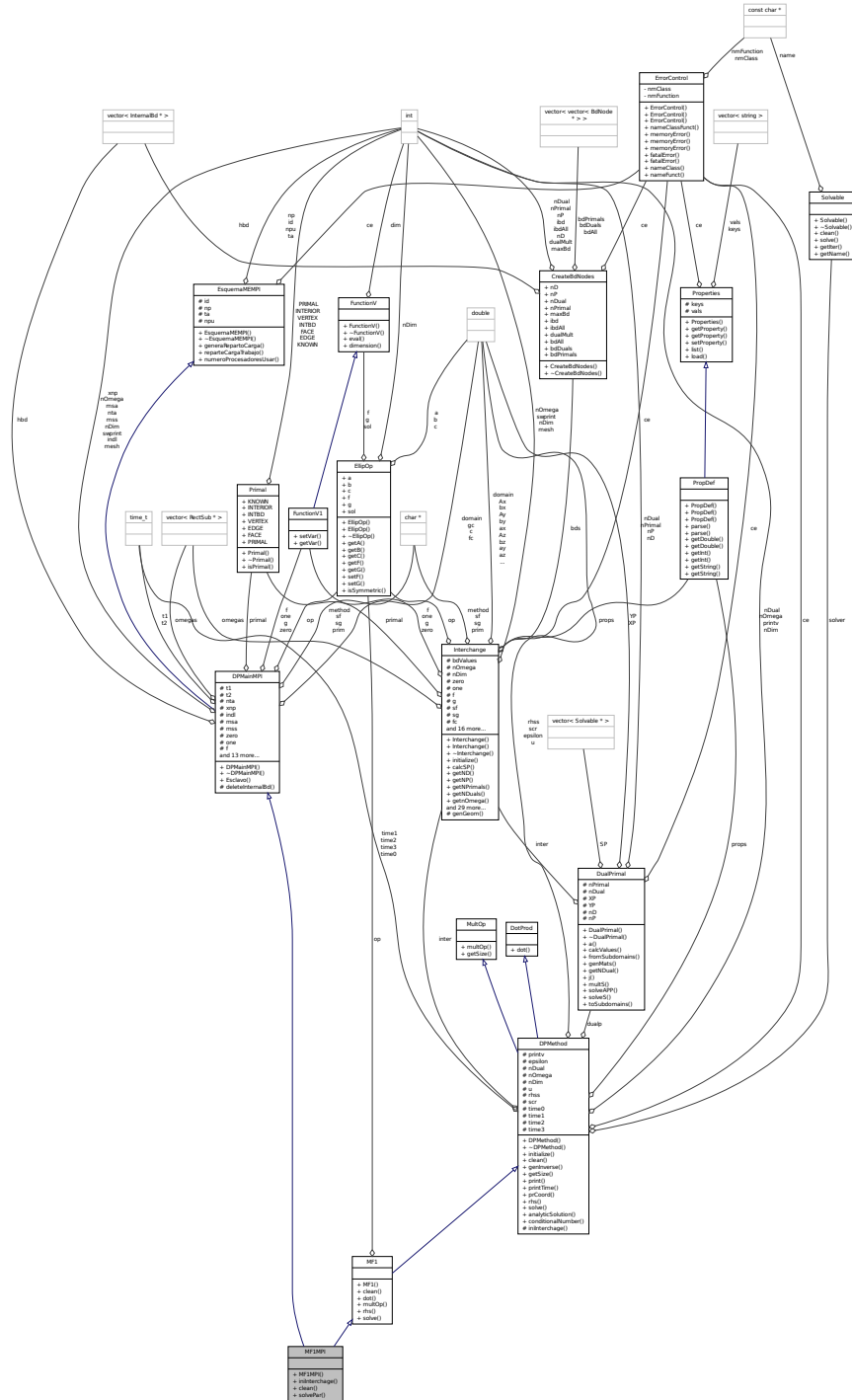
Clase para definir el metodo MF-1 de DVS-DDM.

```
#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 [DPMMethod](#).

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

Sobrecarga del la aplicacion.

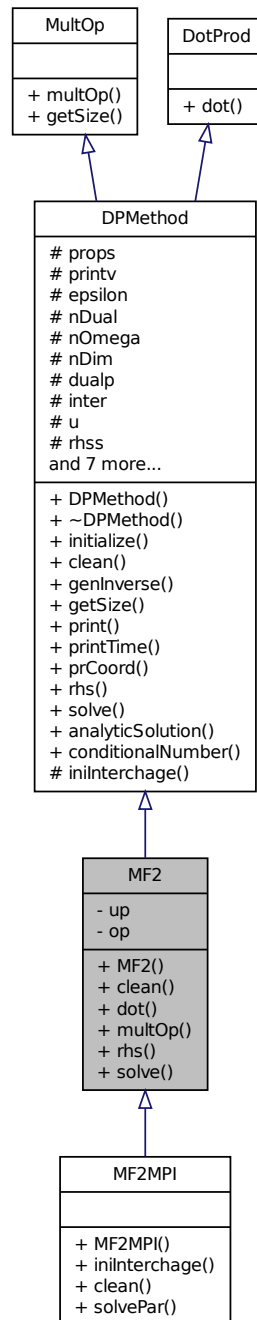
The documentation for this class was generated from the following file:

- [MF1MPI.hpp](#)

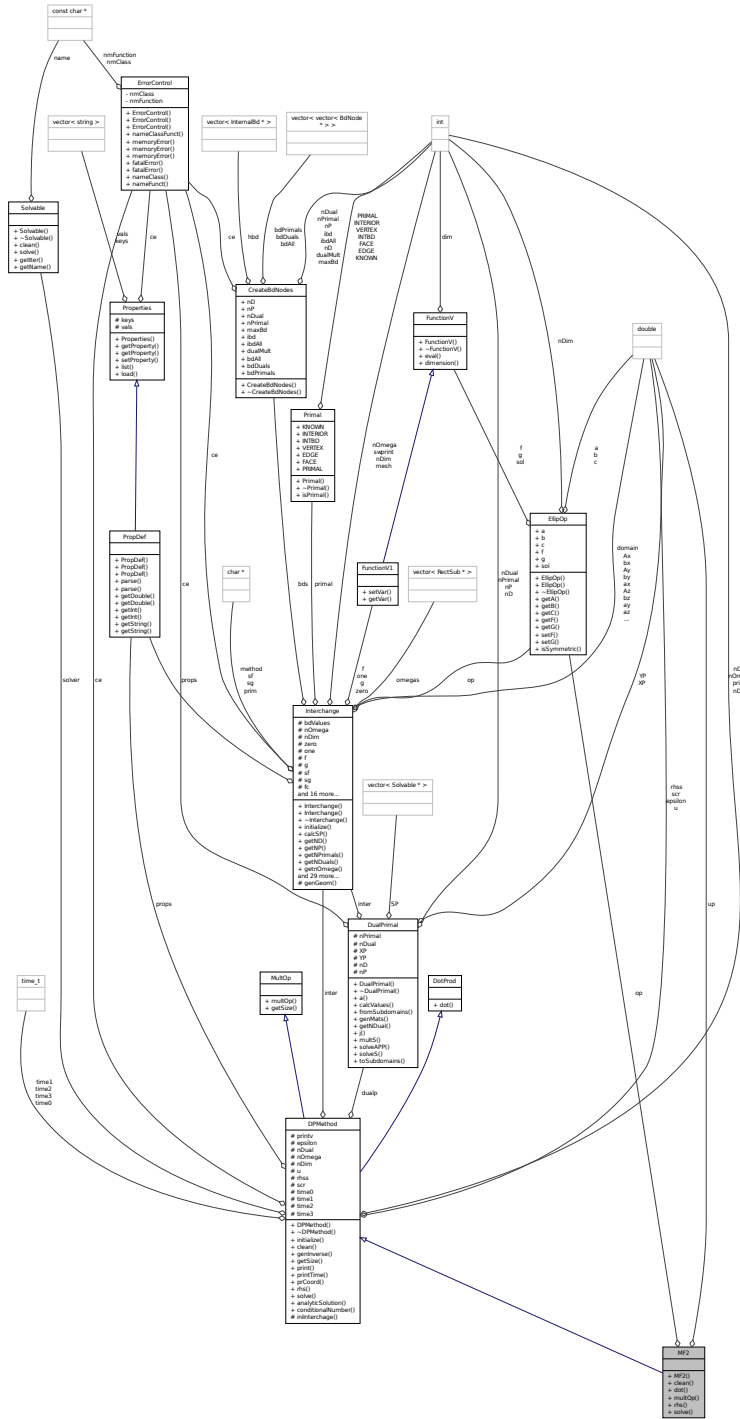
7.43 MF2 Class Reference

```
#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 `Idouble* MF2::up` `[private]`

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

- [MF2.hpp](#)

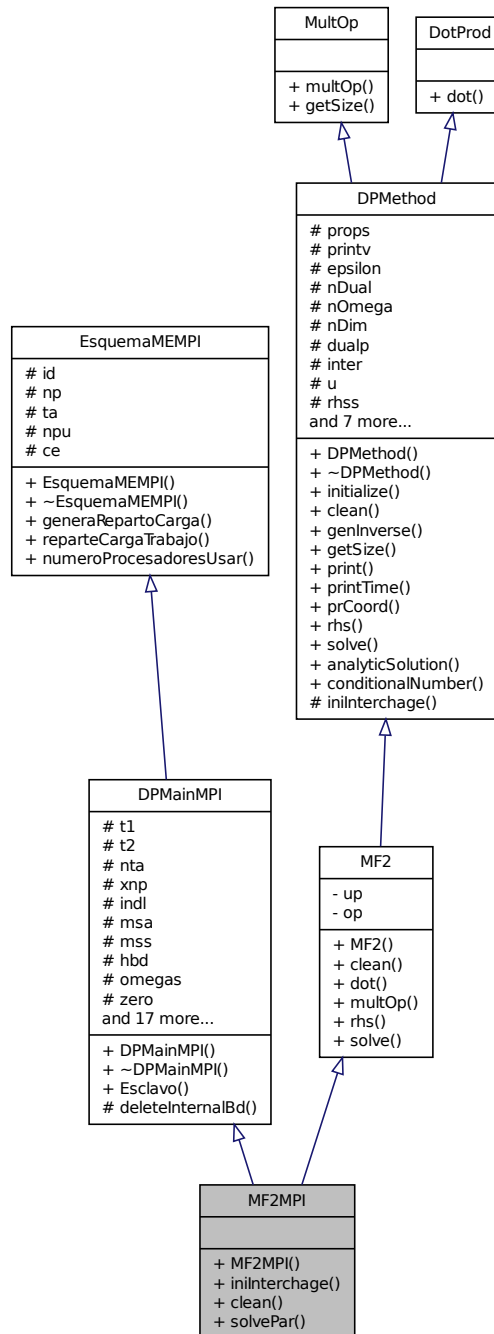
- [MF2.cpp](#)

7.44 MF2MPI Class Reference

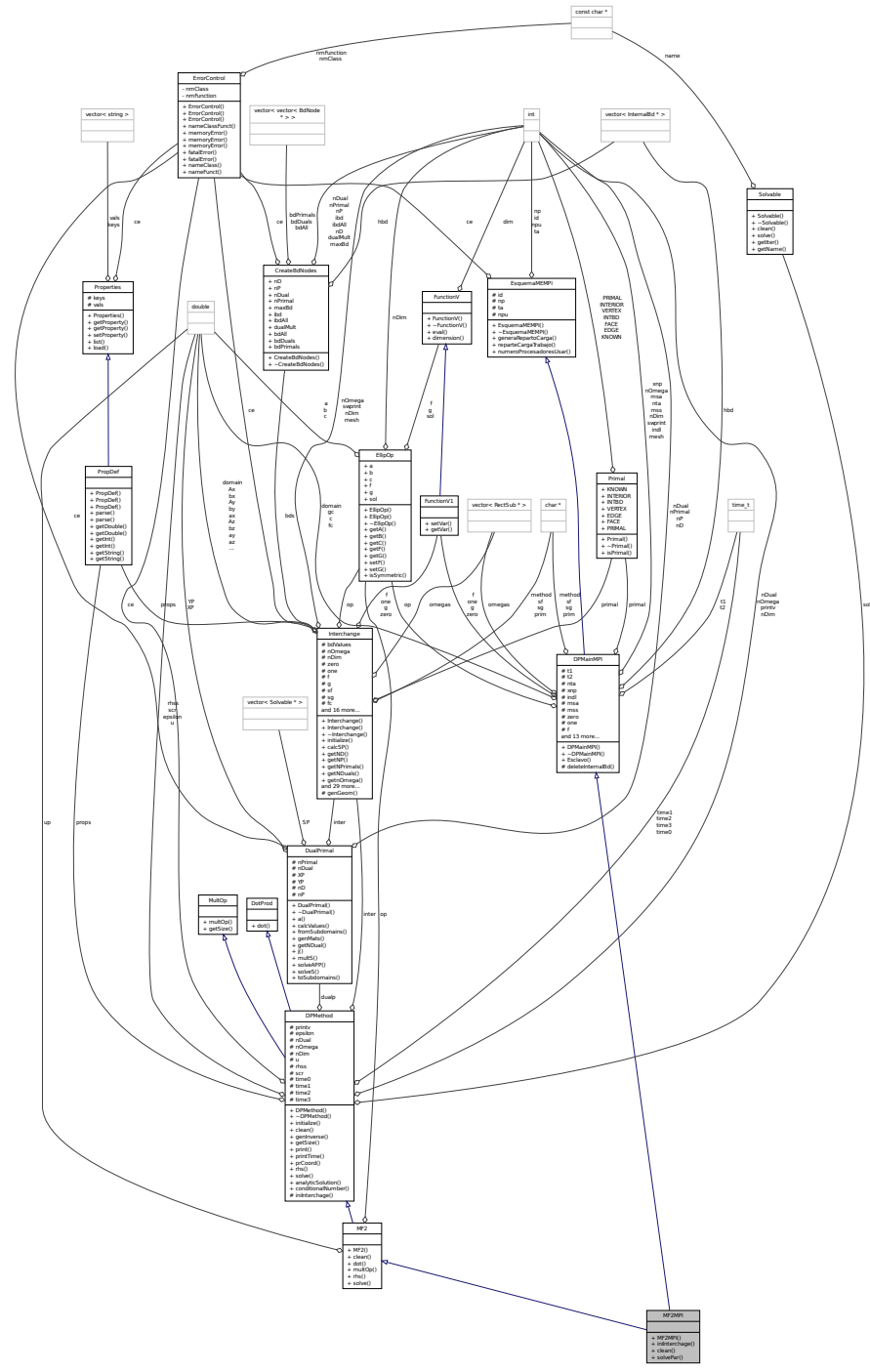
Clase para definir el metodo MF-2 de DVS-DDM.

```
#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.

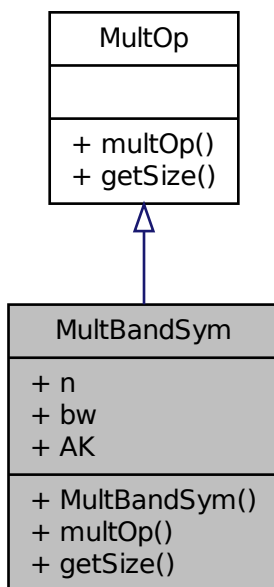
The documentation for this class was generated from the following file:

- [MF2MPI.hpp](#)

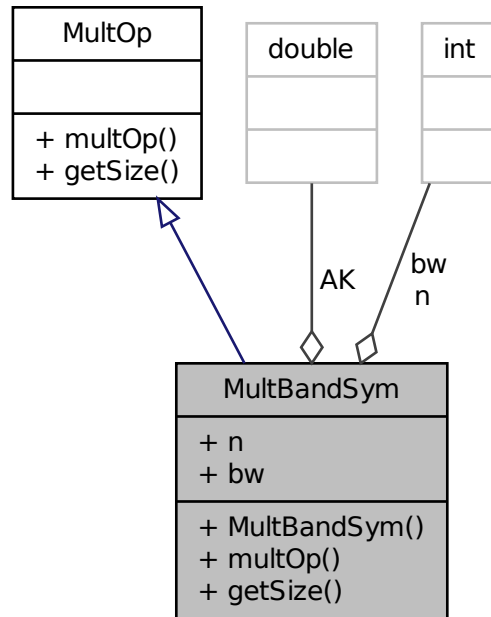
7.45 MultBandSym Class Reference

```
#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`

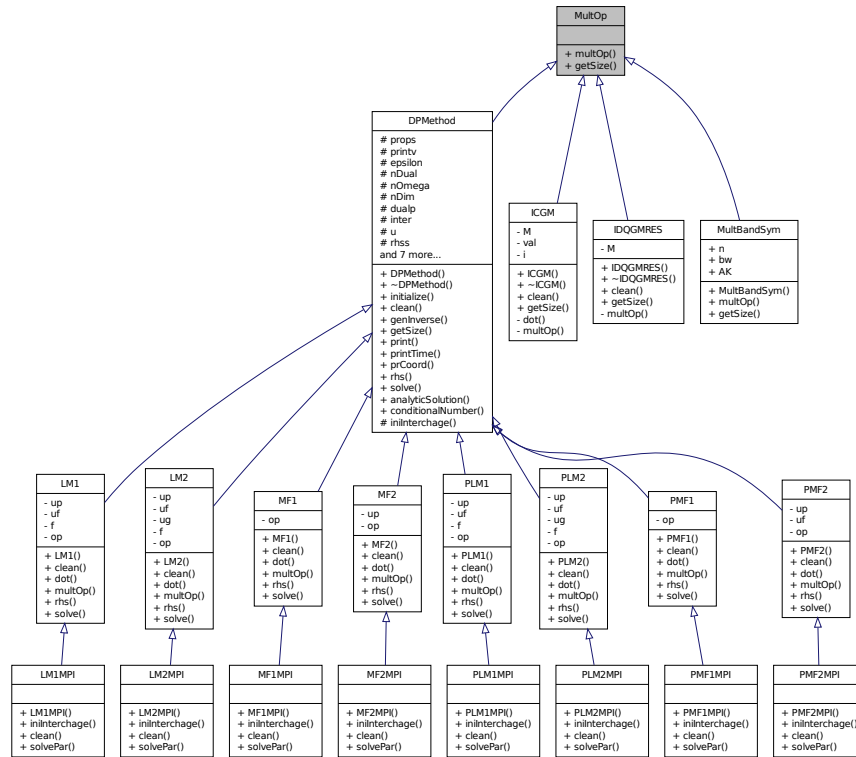
The documentation for this class was generated from the following files:

- [MultBandSym.hpp](#)
- [MultBandSym.cpp](#)

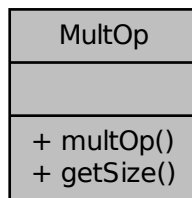
7.46 MultOp Class Reference

```
#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 (Idouble * x, Idouble * y) [pure virtual]`

$y = A*x$

Implemented in [LM2](#), [PLM1](#), [PLM2](#), [LM1](#), [MF2](#), [PMF2](#), [PMF1](#), [MF1](#), and [MultBandSym](#).

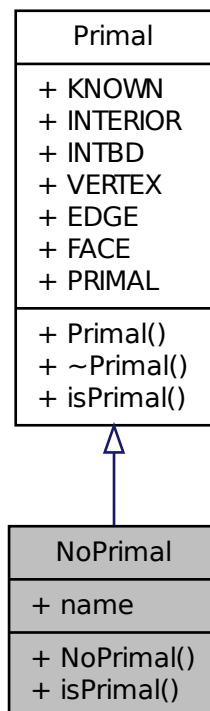
The documentation for this class was generated from the following file:

- [MultOp.hpp](#)

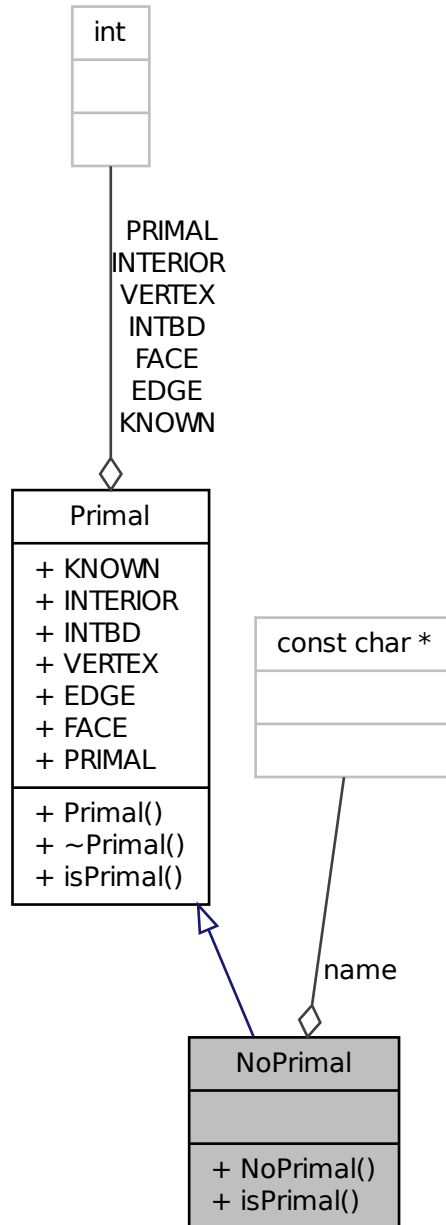
7.47 NoPrimal Class Reference

```
#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`

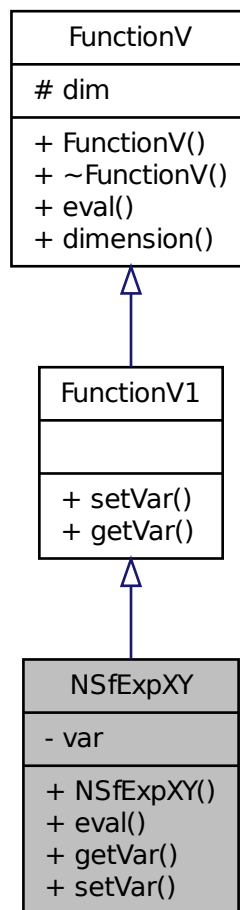
The documentation for this class was generated from the following file:

- [NoPrimal.hpp](#)

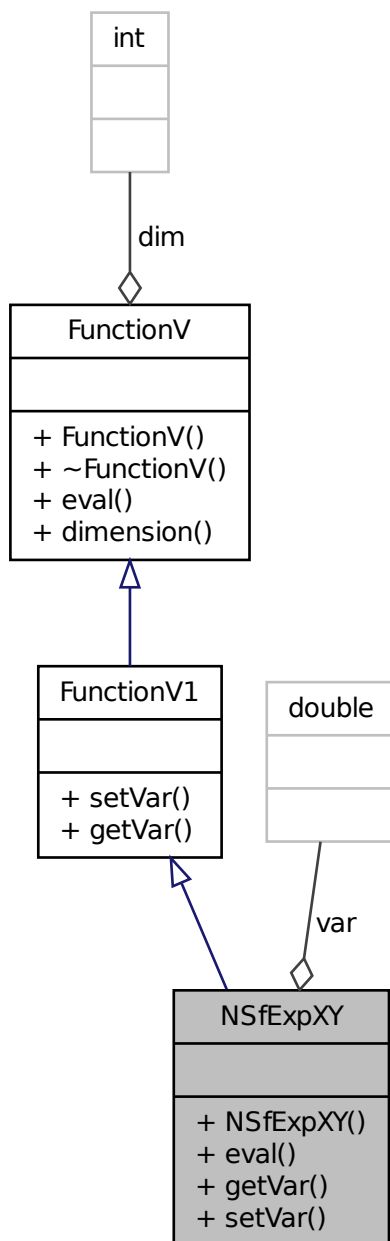
7.48 NSfExpXY Class Reference

```
#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`]

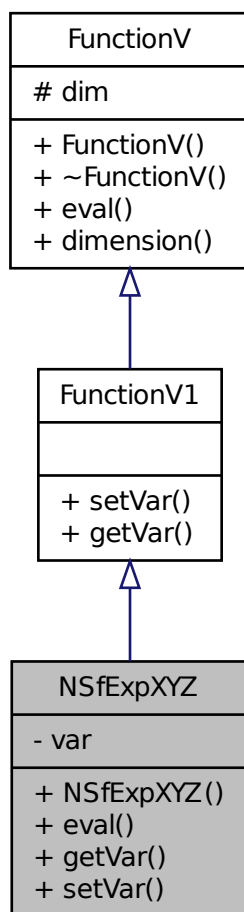
The documentation for this class was generated from the following file:

- [NSfExpXY.hpp](#)

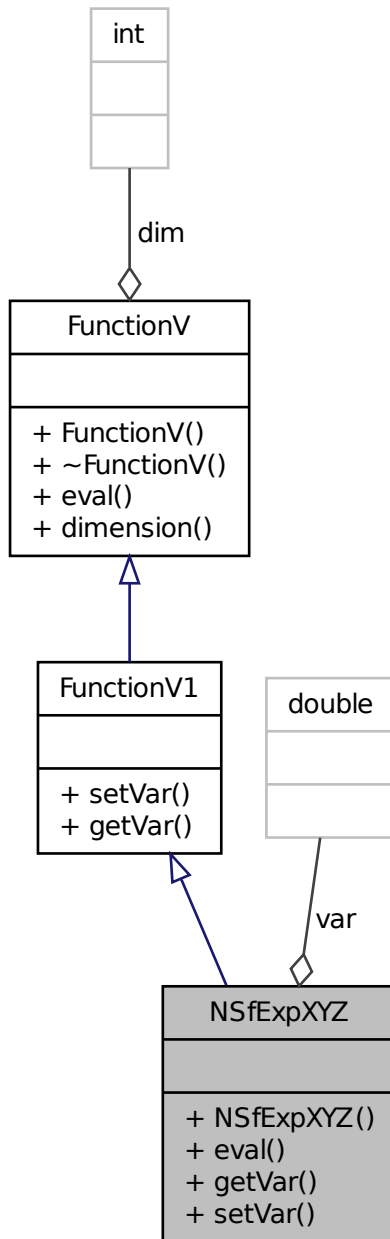
7.49 NSfExpXYZ Class Reference

```
#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

- `ldouble var`

Additional Inherited Members

7.49.1 Constructor & Destructor Documentation

7.49.1.1 `NSfExpXYZ::NSfExpXYZ (ldouble b) [inline]`

7.49.2 Member Function Documentation

7.49.2.1 `ldouble NSfExpXYZ::eval (int d, ldouble * x) [inline],[virtual]`

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

7.49.2.3 `void NSfExpXYZ::setVar (ldouble b) [inline],[virtual]`

Implements [FunctionV1](#).

7.49.3 Member Data Documentation

7.49.3.1 `ldouble NSfExpXYZ::var [private]`

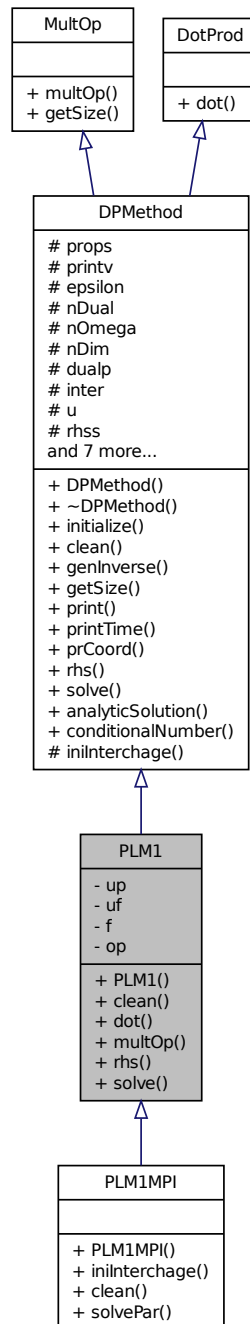
The documentation for this class was generated from the following file:

- [NSfExpXYZ.hpp](#)

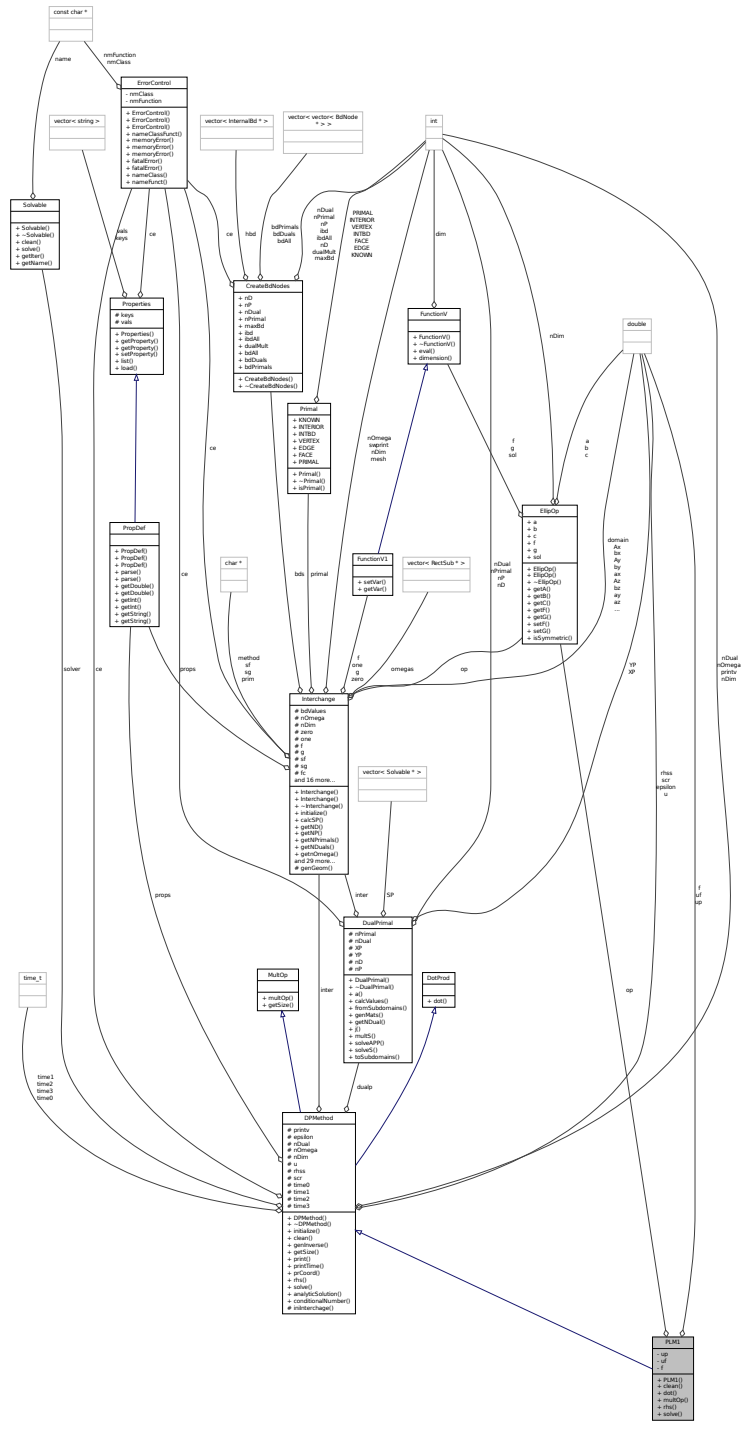
7.50 PLM1 Class Reference

```
#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 `Idouble* PLM1::f` [private]

7.50.3.2 `EllipOp* PLM1::op` [private]

7.50.3.3 `Idouble* PLM1::uf` [private]

7.50.3.4 `Idouble* PLM1::up` [private]

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

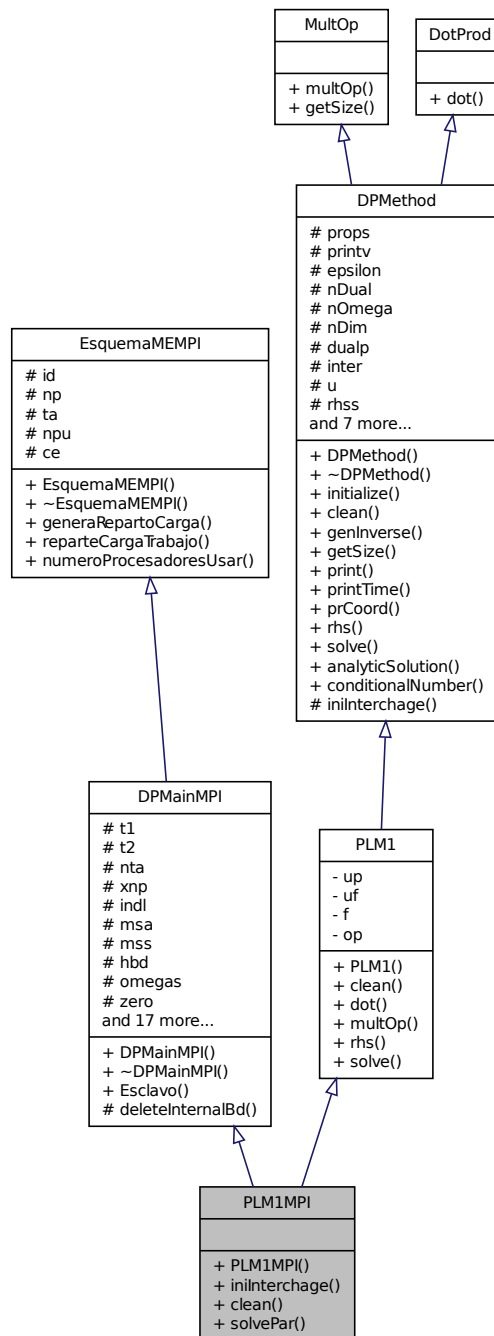
- [PLM1.hpp](#)
- [PLM1.cpp](#)

7.51 PLM1MPI Class Reference

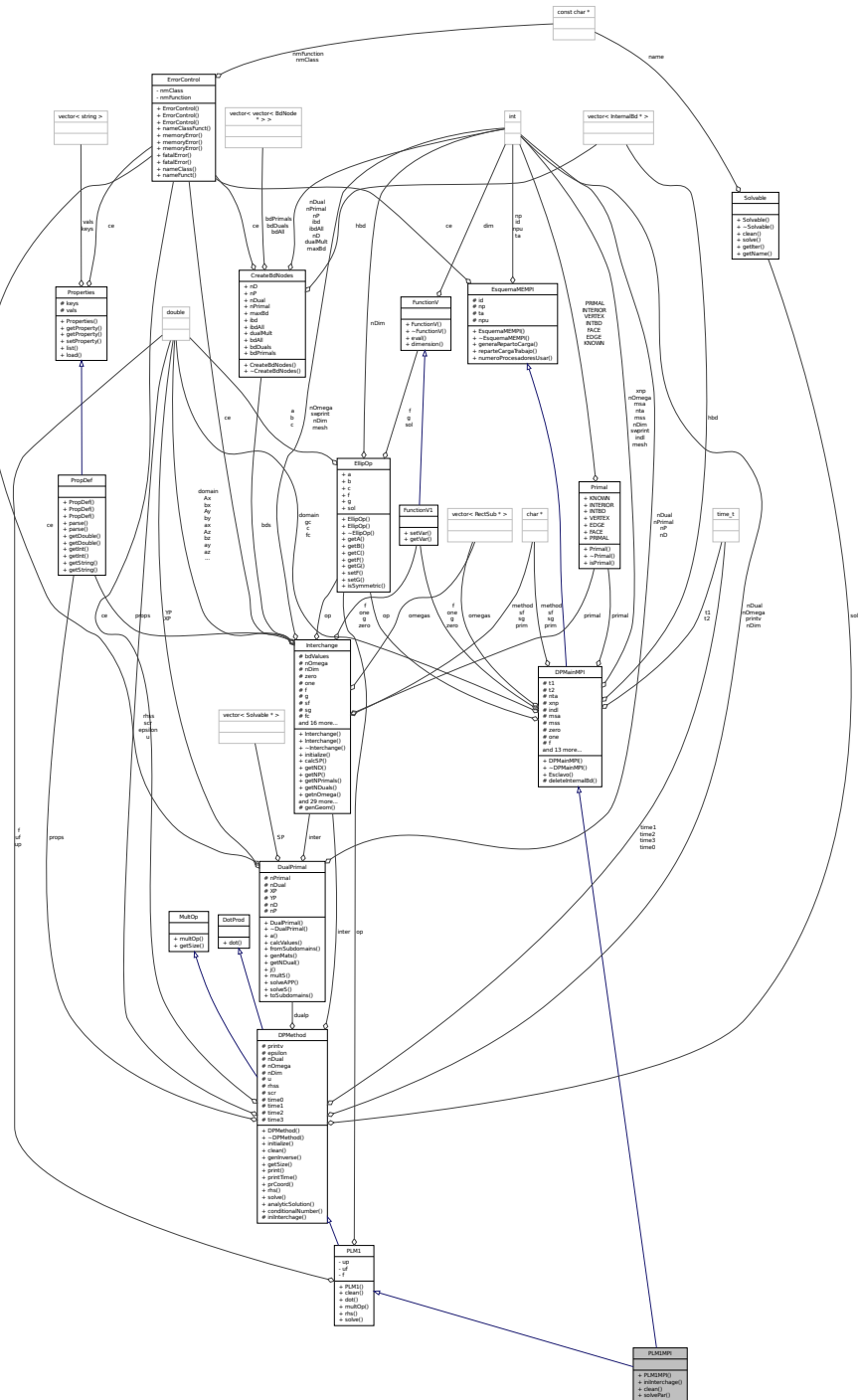
Clase para definir el metodo PLM-1 de DVS-DDM.

```
#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.

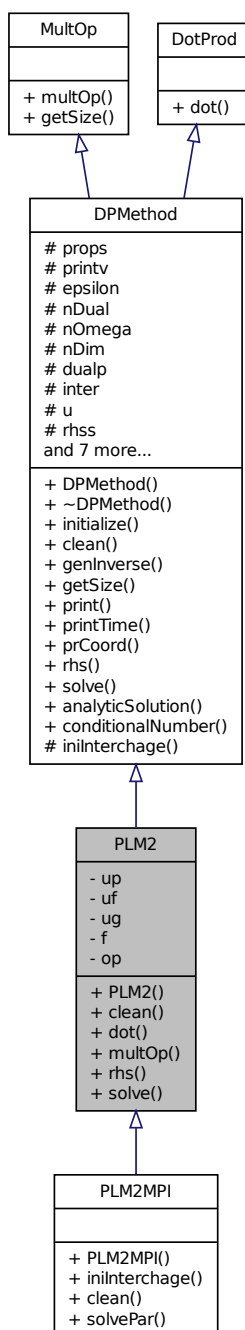
The documentation for this class was generated from the following file:

- [PLM1MPI.hpp](#)

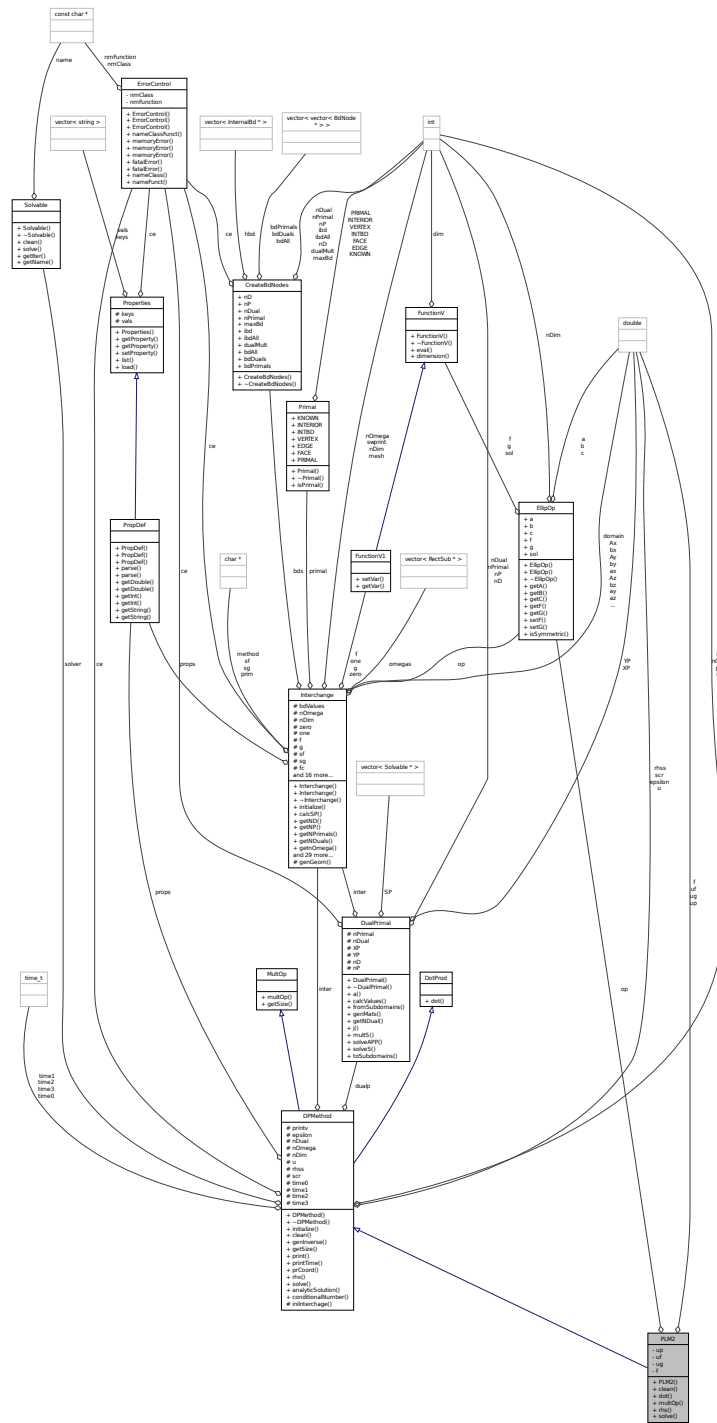
7.52 PLM2 Class Reference

```
#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 `Idouble* PLM2::f` [private]

7.52.3.2 `EllipOp* PLM2::op` [private]

7.52.3.3 `Idouble* 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](#)

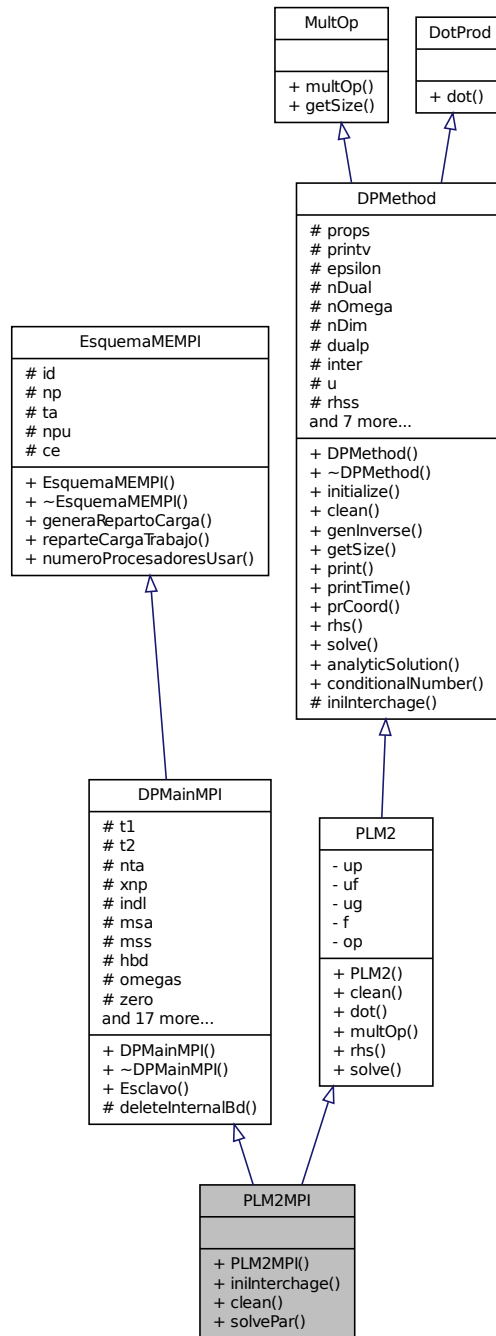
- [PLM2.cpp](#)

7.53 PLM2MPI Class Reference

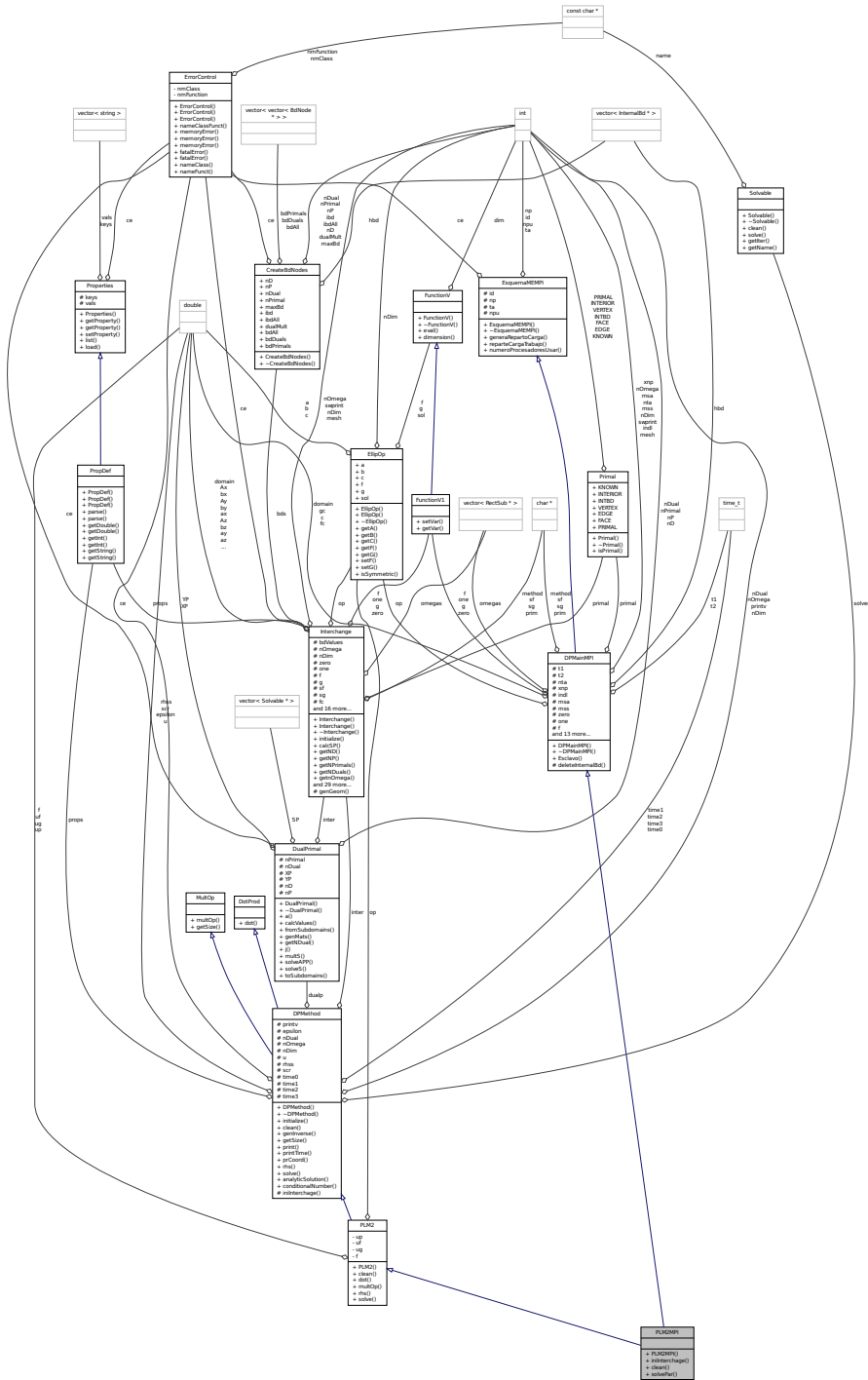
Clase para definir el metodo MF-1 de DVS-DDM.

```
#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.

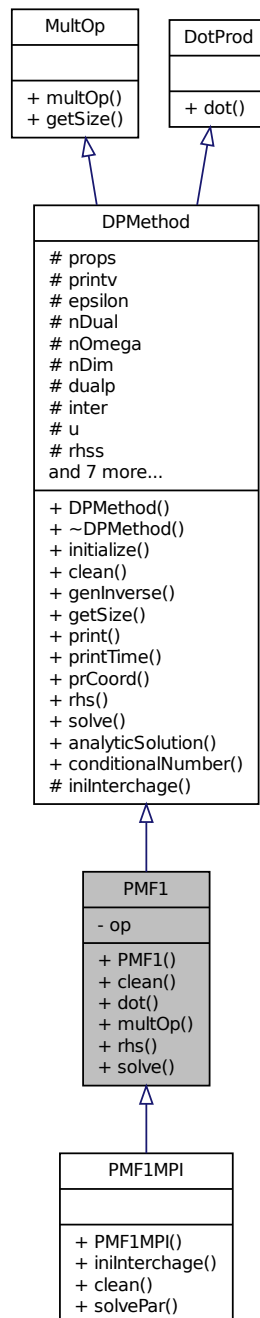
The documentation for this class was generated from the following file:

- [PLM2MPI.hpp](#)

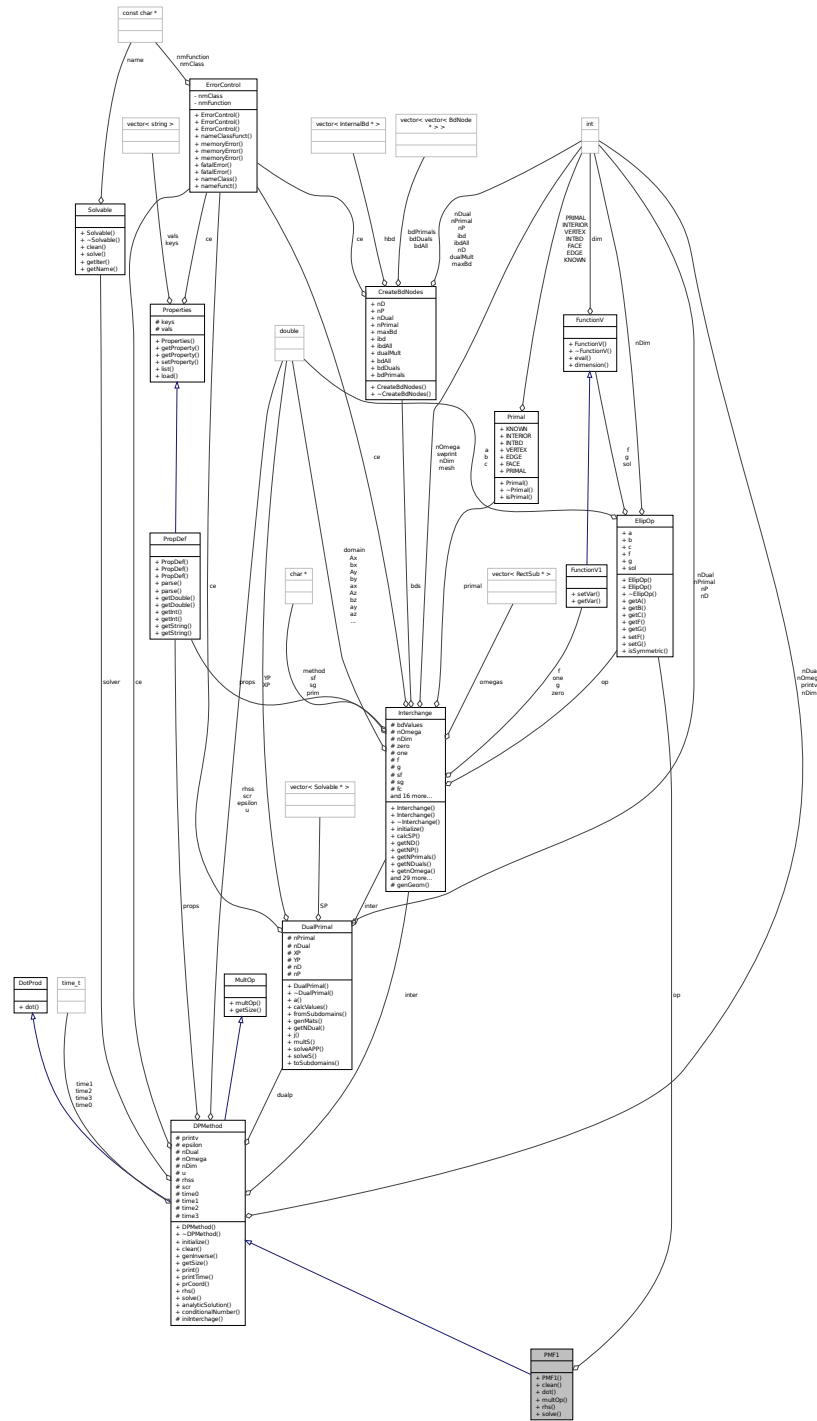
7.54 PMF1 Class Reference

```
#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](#)

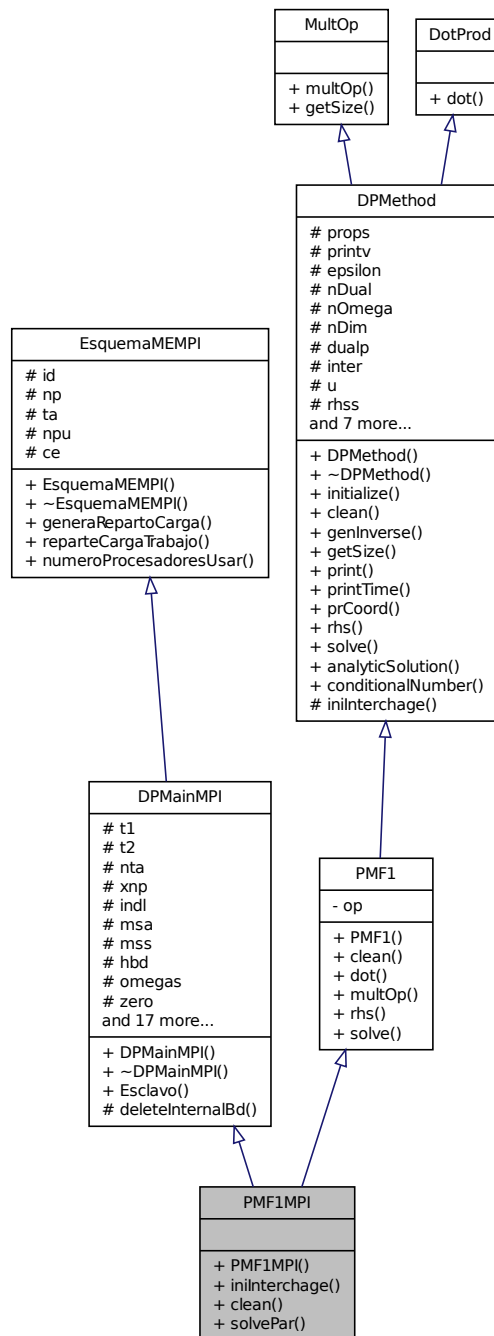
- [PMF1.cpp](#)

7.55 PMF1MPI Class Reference

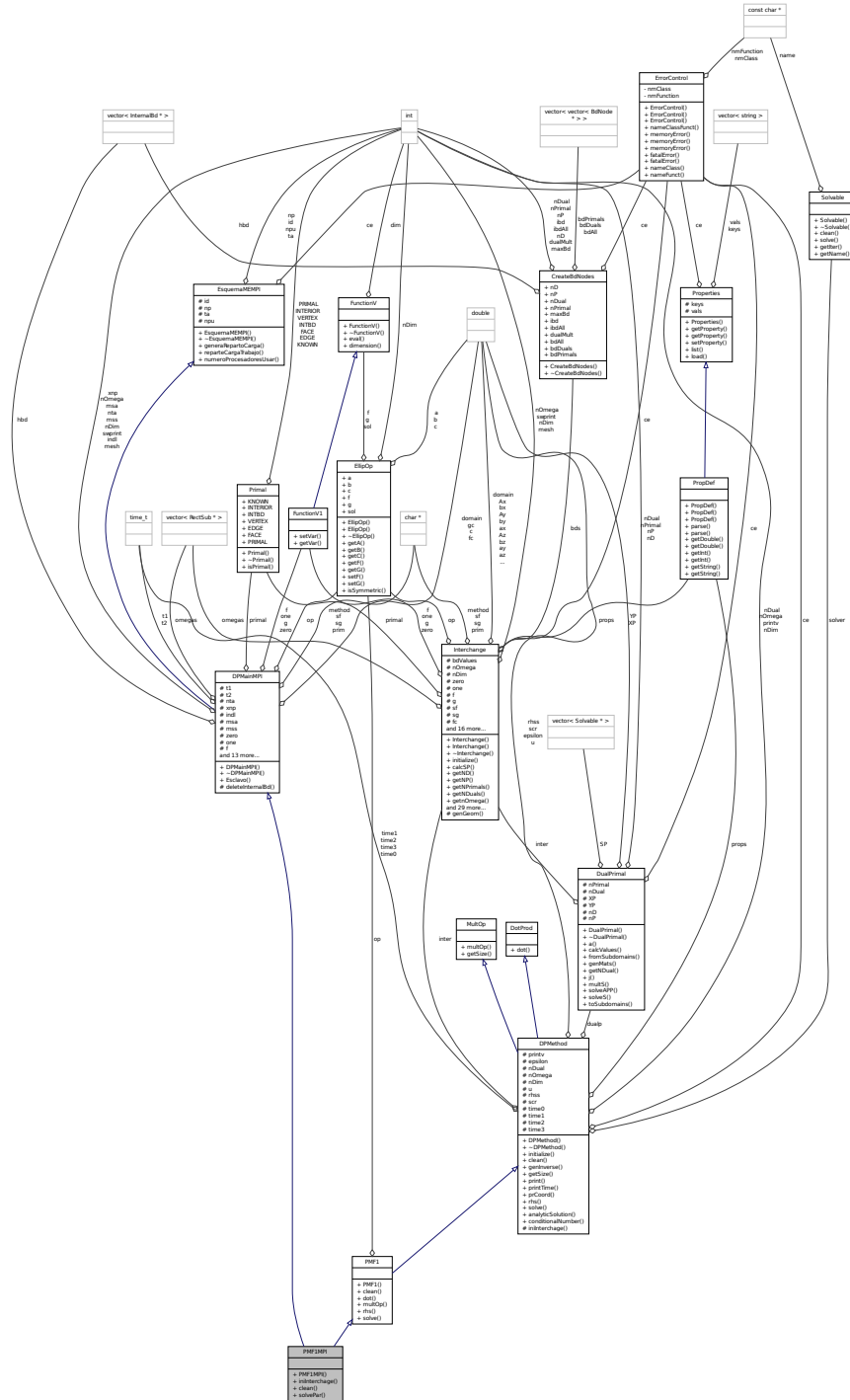
Clase para definir el metodo PMF-1 de DVS-DDM.

```
#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 [DPMMethod](#).

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

Sobrecarga del la aplicacion.

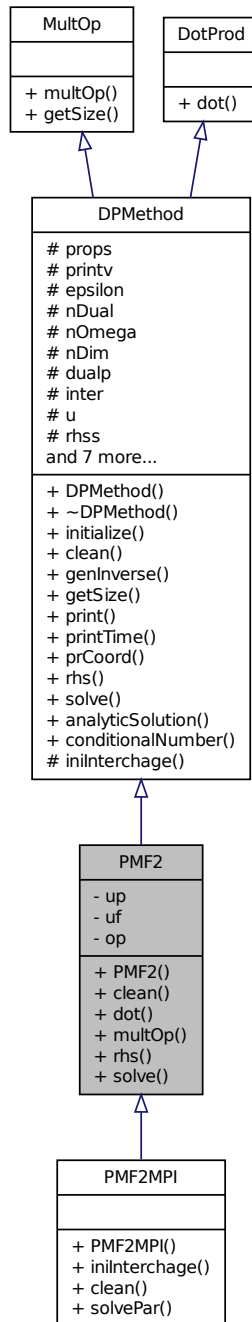
The documentation for this class was generated from the following file:

- [PMF1MPI.hpp](#)

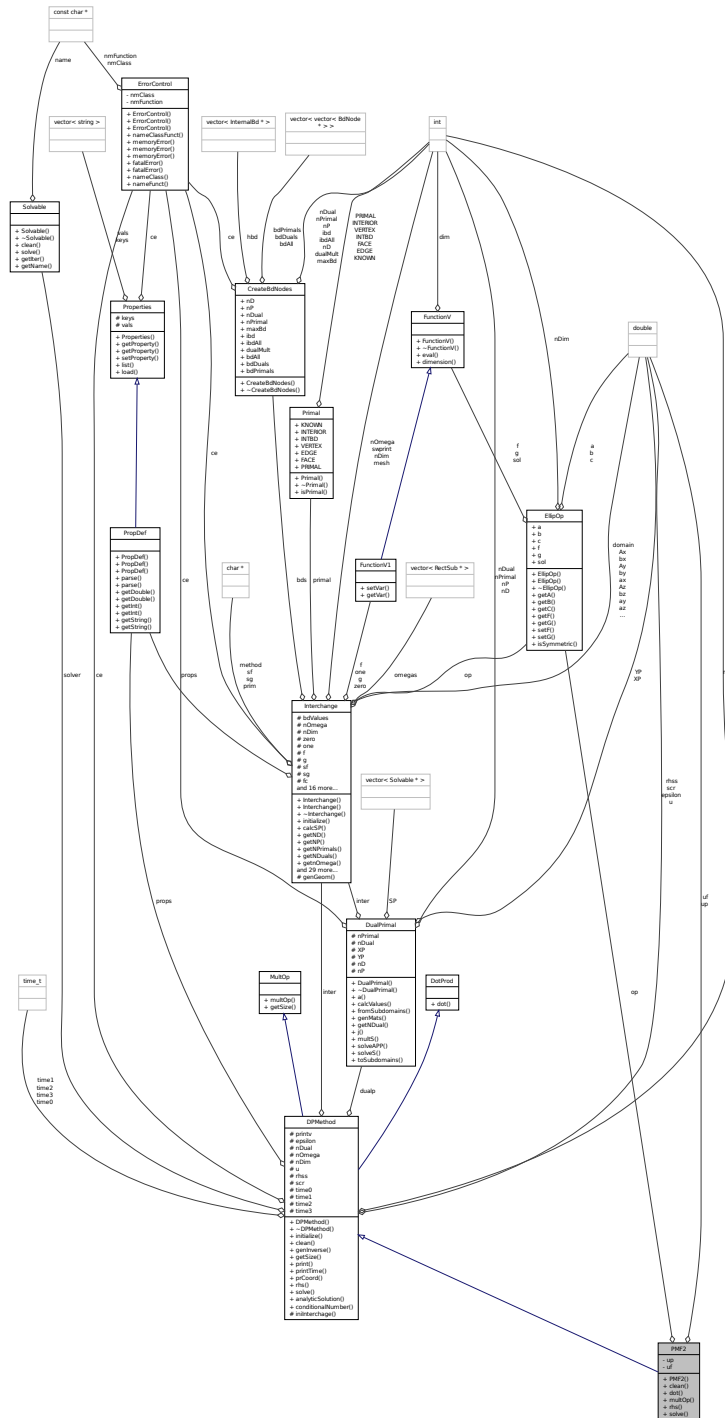
7.56 PMF2 Class Reference

```
#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 `Idouble* PMF2::uf` [private]

7.56.3.3 `Idouble* PMF2::up` [private]

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

- [PMF2.hpp](#)

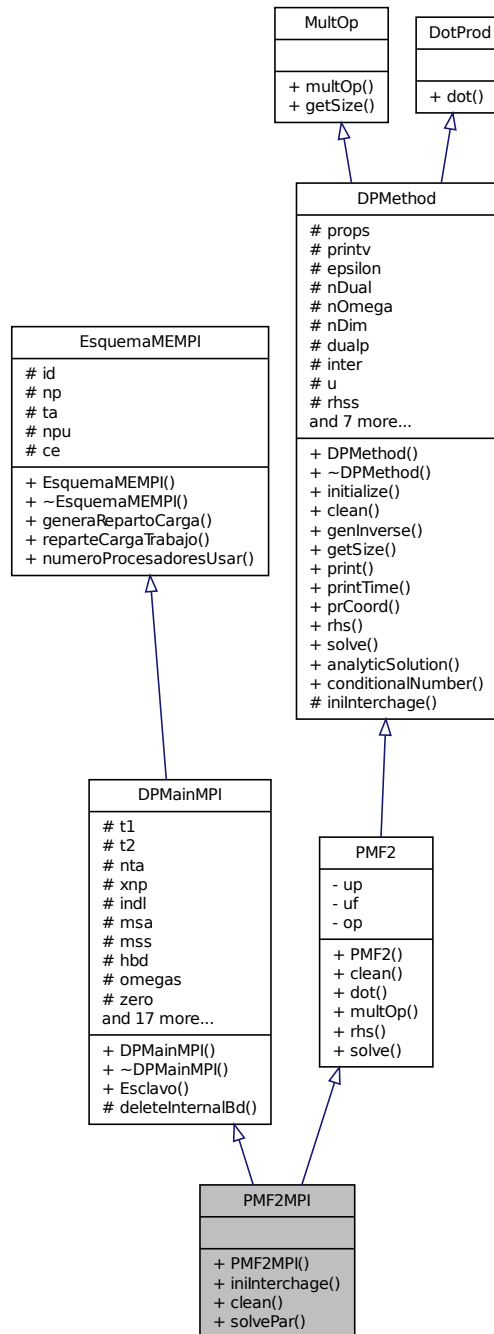
- [PMF2.cpp](#)

7.57 PMF2MPI Class Reference

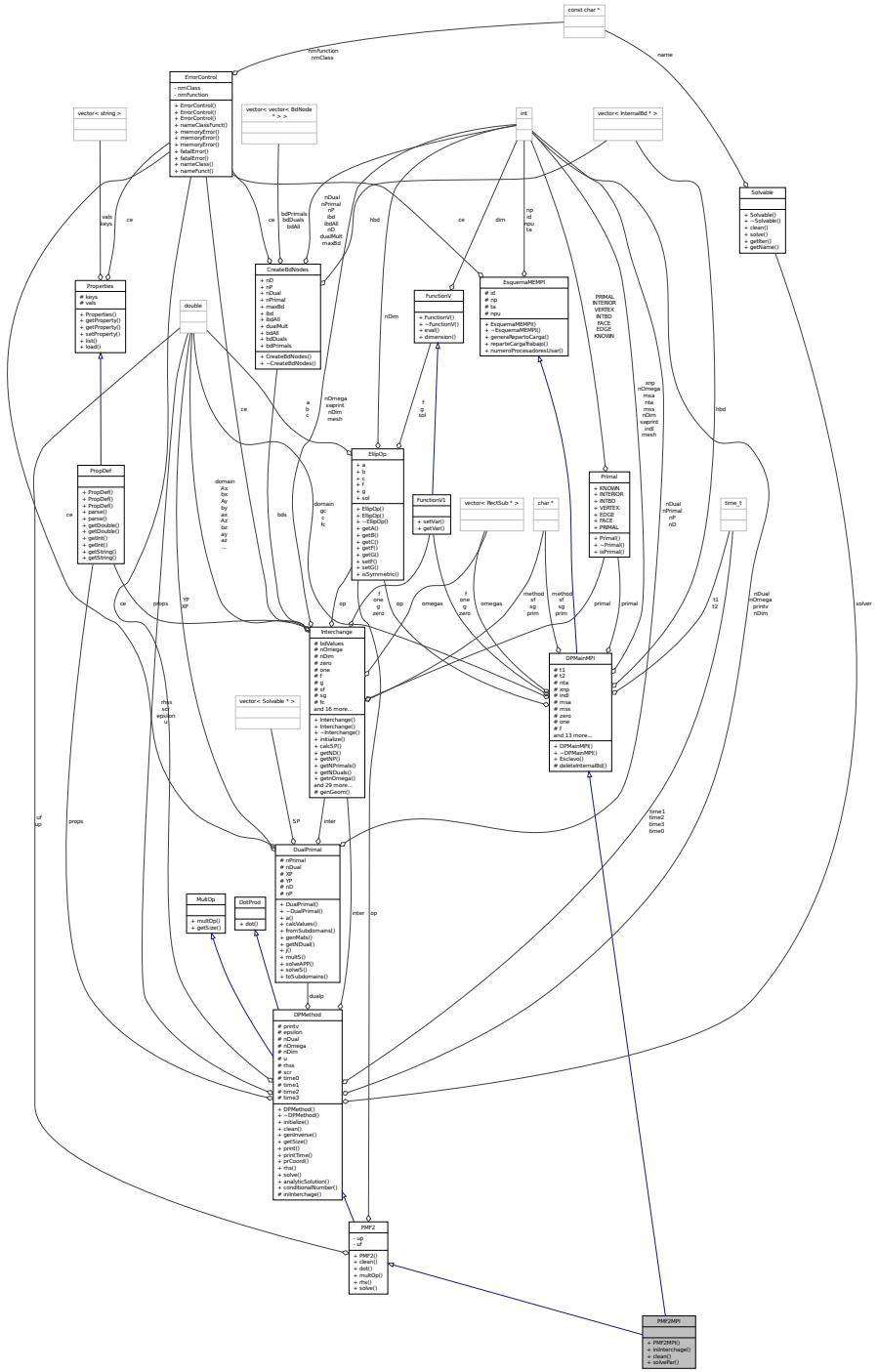
Clase para definir el metodo PMF-2 de DVS-DDM.

```
#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.

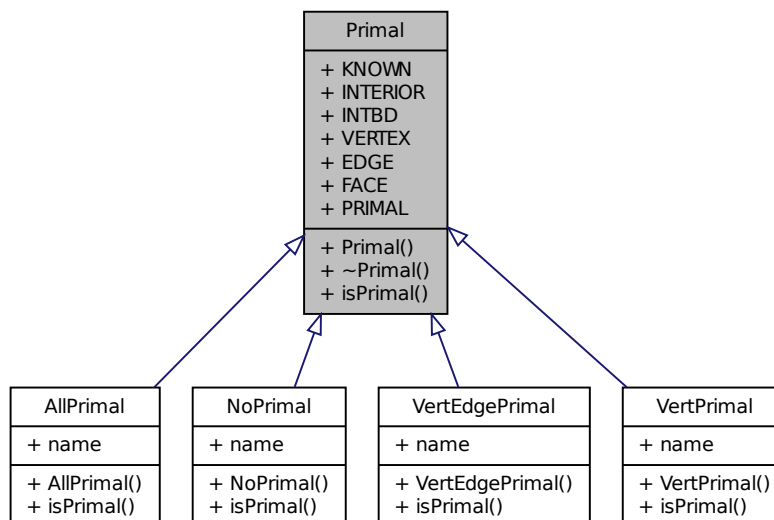
The documentation for this class was generated from the following file:

- [PMF2MPI.hpp](#)

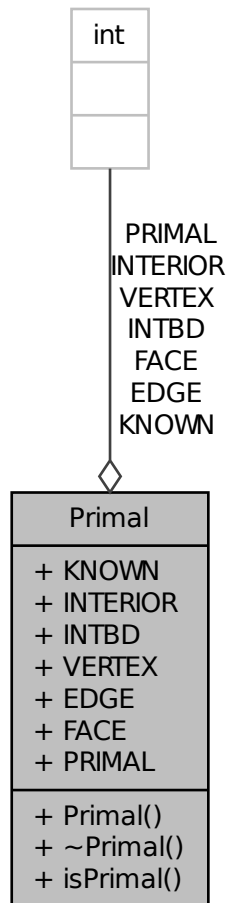
7.58 Primal Class Reference

```
#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`]

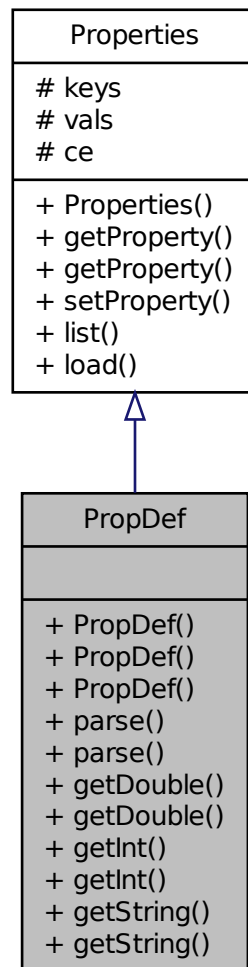
The documentation for this class was generated from the following file:

- [Primal.hpp](#)

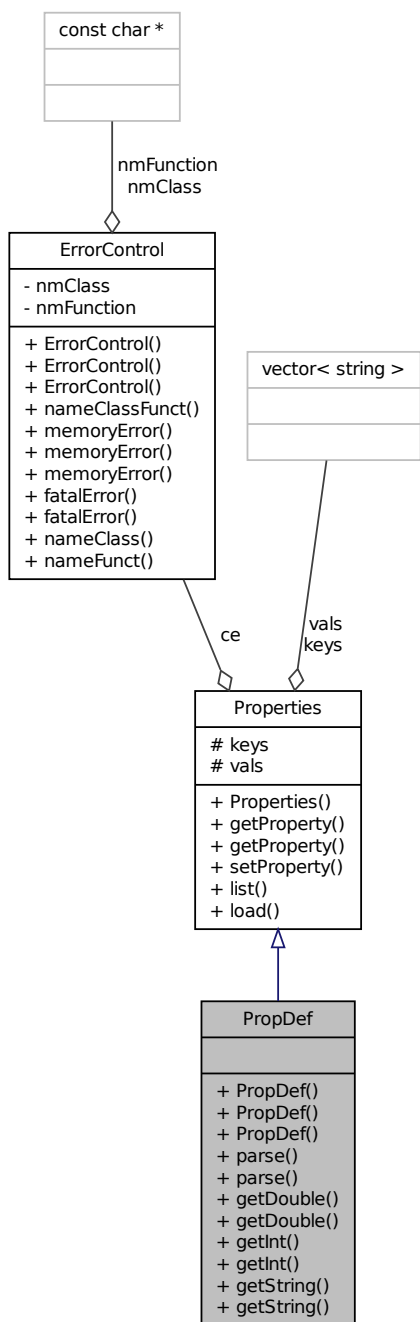
7.59 PropDef Class Reference

```
#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[])`

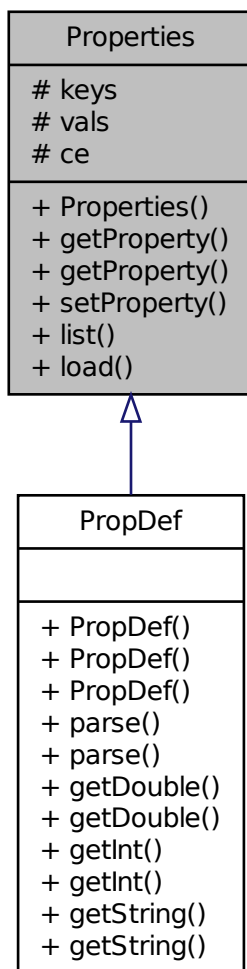
The documentation for this class was generated from the following files:

- [PropDef.hpp](#)
- [PropDef.cpp](#)

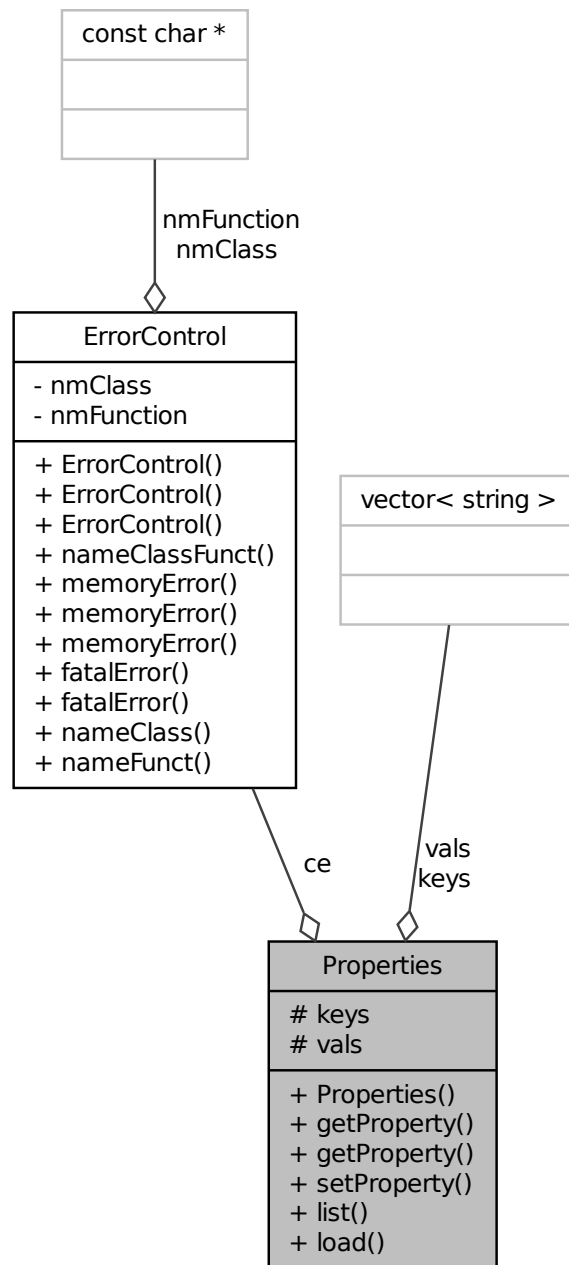
7.60 Properties Class Reference

```
#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]`

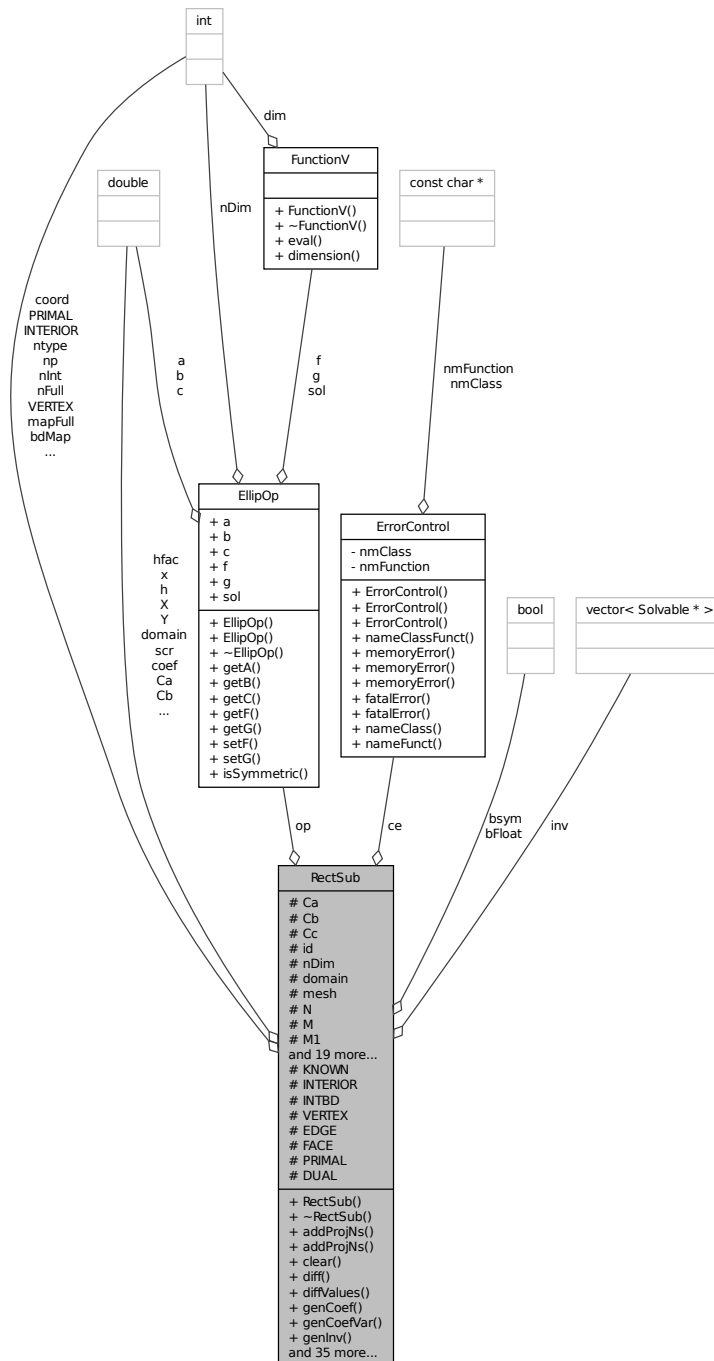
The documentation for this class was generated from the following files:

- [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)

- int `addProjNs` (`ldouble **A`, int `*map`, `ldouble` `fac`)
- int `addProjNs` (`MatrizDispersa *A`, int `*map`, `ldouble` `fac`)
- 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`)
- `Solvable * genInverse` (int `*map`, `ldouble` `fac`)
- 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)
- `vector< Solvable * > getInv` (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](#)
- [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](#)
- [vector< Solvable * > inv](#)
- [ldouble * coef](#)
- [int * bdMap](#)
- [int * mapInt](#)
- [int * mapFull](#)
- [int nInt](#)
- [int nFull](#)
- [ldouble * X](#)
- [ldouble * Y](#)
- [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, Idouble ** dom, EllipOp & op, Primal & primal)`

7.61.1.2 `RectSub::~~RectSub (void)` `[inline]`

7.61.2 Member Function Documentation

7.61.2.1 `int RectSub::addProjNs (Idouble ** A, int * map, Idouble fac)`

7.61.2.2 `int RectSub::addProjNs (MatrizDispersa * A, int * map, Idouble fac)`

7.61.2.3 `void RectSub::clear (int s)`

7.61.2.4 `void RectSub::diff (int sc3, int sc1, int sc2)`

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

7.61.2.6 `void RectSub::genCoef (EllipOp & op)`

7.61.2.7 `void RectSub::genCoefVar (int ren)`

7.61.2.8 `void RectSub::genInv (int type)`

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

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

7.61.2.11 `void RectSub::genNtype (Primal & primal)`

7.61.2.12 `int RectSub::getBdSize ()` `[inline]`

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

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

7.61.2.15 `vector< InternalBd * > RectSub::getInternalBd (void)`

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

7.61.2.17 `int RectSub::getNP (void)` `[inline]`

7.61.2.18 `int* RectSub::getNtype (void)` `[inline]`

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

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

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

7.61.2.22 `void RectSub::inverse (int sp, int sc1, int sc2)`

- 7.61.2.23 `bool RectSub::isDual (int i)` `[inline]`
- 7.61.2.24 `bool RectSub::isFloat (void)` `[inline]`
- 7.61.2.25 `bool RectSub::isIntBd (int * coord)`
- 7.61.2.26 `bool RectSub::isInterior (int * coord)`
- 7.61.2.27 `bool RectSub::isInterior (int i)` `[inline]`
- 7.61.2.28 `bool RectSub::isKnown (int * coord)`
- 7.61.2.29 `bool RectSub::isKnown (int i)` `[inline]`
- 7.61.2.30 `bool RectSub::isPrimal (int i)` `[inline]`
- 7.61.2.31 `bool RectSub::isVertex (int i)` `[inline]`
- 7.61.2.32 `void RectSub::knownValues (int s1)`
- 7.61.2.33 `void RectSub::multOp (int s1, int s2)`
- 7.61.2.34 `int RectSub::nodeType (int * coord)`
- 7.61.2.35 `void RectSub::print (const char * s, int sc)`
- 7.61.2.36 `void RectSub::print (int sc)`
- 7.61.2.37 `void RectSub::printMat (const char * s, Idouble ** A, int tm)`
- 7.61.2.38 `void RectSub::printMult (void)`
- 7.61.2.39 `void RectSub::rhs (int sc)`
- 7.61.2.40 `void RectSub::setNtype (int * arr)` `[inline]`
- 7.61.2.41 `void RectSub::setPrimals (int sc, Idouble * u)`
- 7.61.2.42 `void RectSub::setValue (int sc, int n, Idouble val)` `[inline]`
- 7.61.2.43 `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]

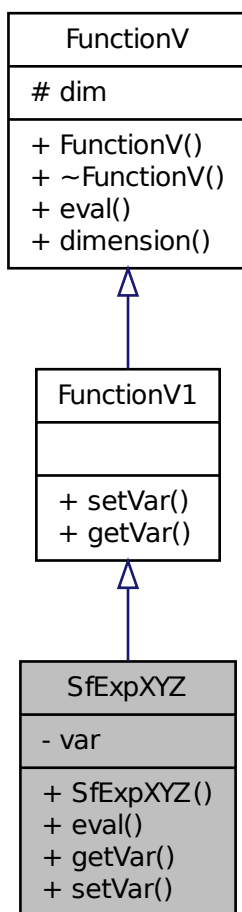
The documentation for this class was generated from the following files:

- [RectSub.hpp](#)
- [RectSub.cpp](#)

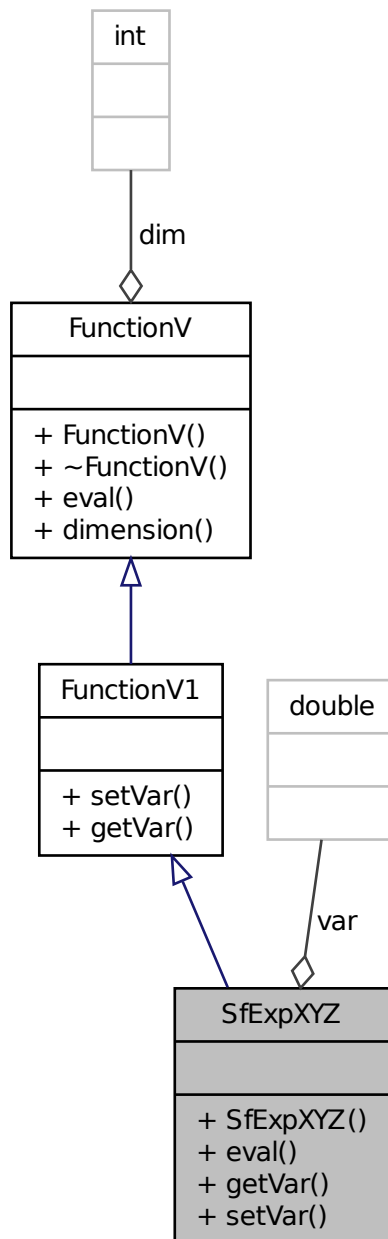
7.62 SfExpXYZ Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.62.1 Constructor & Destructor Documentation

7.62.1.1 [SfExpXYZ::SfExpXYZ \(ldouble b \)](#) [[inline](#)]

7.62.2 Member Function Documentation

7.62.2.1 [ldouble SfExpXYZ::eval \(int d, ldouble * x \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV](#).

7.62.2.2 [ldouble SfExpXYZ::getVar \(void \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.62.2.3 [void SfExpXYZ::setVar \(ldouble b \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.62.3 Member Data Documentation

7.62.3.1 [ldouble SfExpXYZ::var](#) [[private](#)]

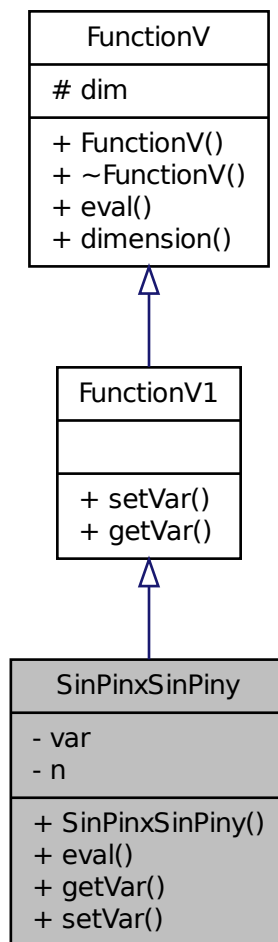
The documentation for this class was generated from the following file:

- [SfExpXYZ.hpp](#)

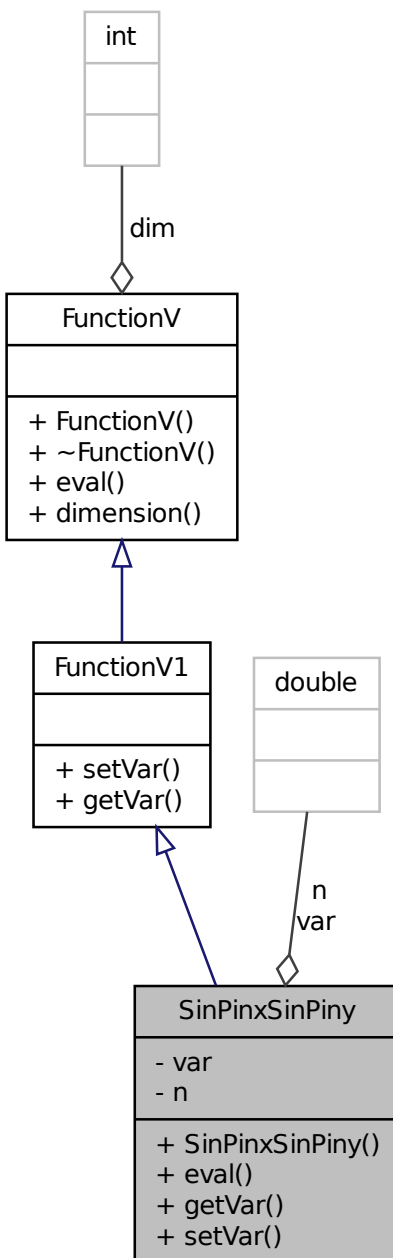
7.63 SinPinxSinPiny Class Reference

```
#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]

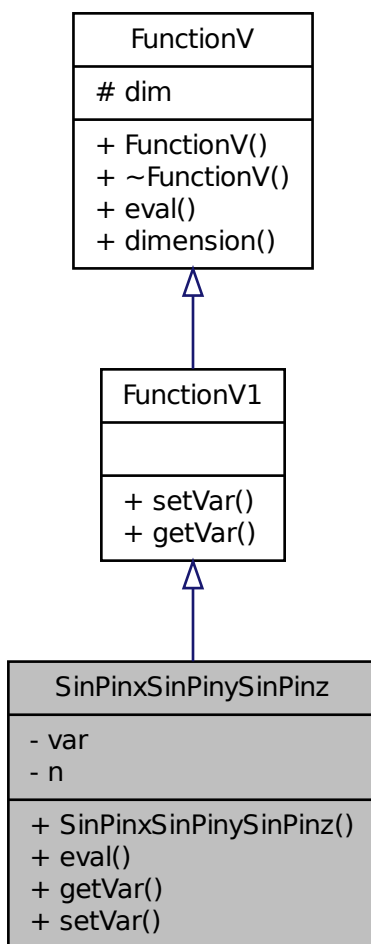
The documentation for this class was generated from the following file:

- [SinPinxSinPiny.hpp](#)

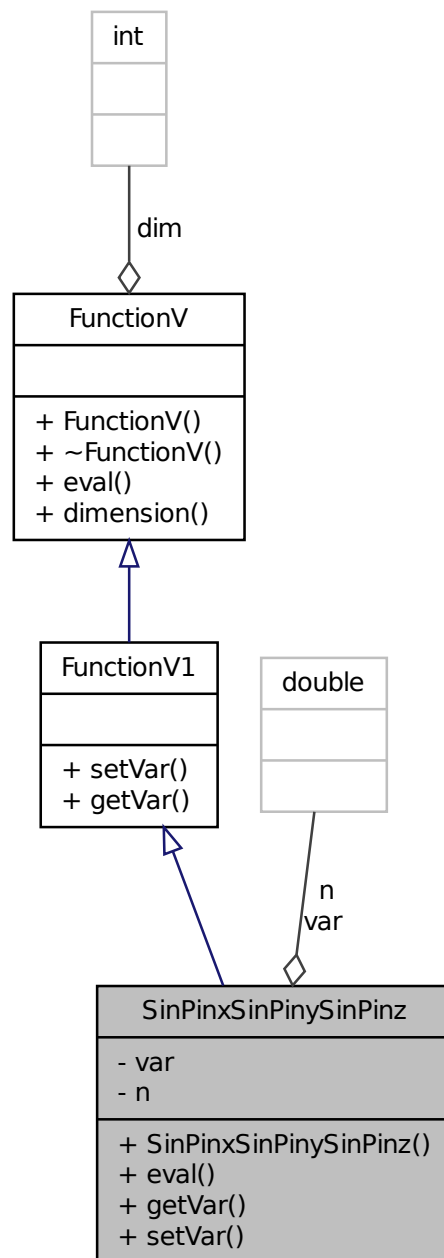
7.64 SinPinxSinPinySinPinz Class Reference

```
#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 [SinPixSinPinySinPinz::SinPixSinPinySinPinz \(\[ldouble\]\(#\) b \)](#) [[inline](#)]

7.64.2 Member Function Documentation

7.64.2.1 [ldouble SinPixSinPinySinPinz::eval \(\[int\]\(#\) d, \[ldouble\]\(#\) * x \)](#) [[inline](#)], [[virtual](#)]

Implements [FunctionV](#).

7.64.2.2 [ldouble SinPixSinPinySinPinz::getVar \(\[void\]\(#\) \)](#) [[inline](#)], [[virtual](#)]

Implements [FunctionV1](#).

7.64.2.3 [void SinPixSinPinySinPinz::setVar \(\[ldouble\]\(#\) b \)](#) [[inline](#)], [[virtual](#)]

Implements [FunctionV1](#).

7.64.3 Member Data Documentation

7.64.3.1 [ldouble SinPixSinPinySinPinz::n](#) [[private](#)]

7.64.3.2 [ldouble SinPixSinPinySinPinz::var](#) [[private](#)]

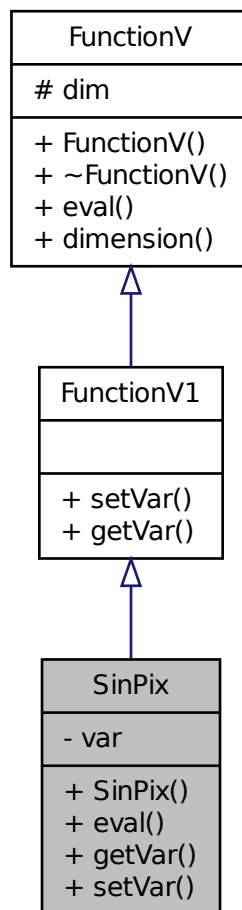
The documentation for this class was generated from the following file:

- [SinPixSinPinySinPinz.hpp](#)

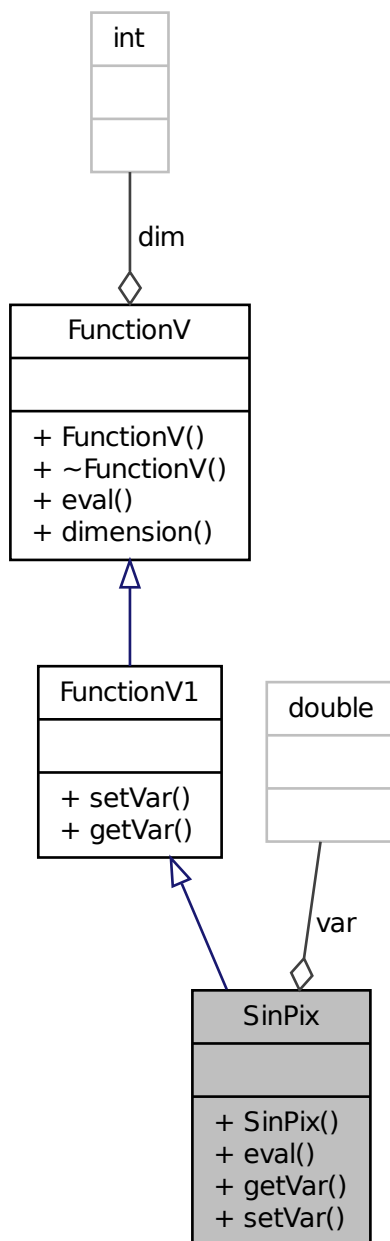
7.65 SinPix Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.65.1 Constructor & Destructor Documentation

7.65.1.1 [SinPix::SinPix \(ldouble b \)](#) [[inline](#)]

7.65.2 Member Function Documentation

7.65.2.1 [ldouble SinPix::eval \(int d, ldouble * x \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV](#).

7.65.2.2 [ldouble SinPix::getVar \(void \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.65.2.3 [void SinPix::setVar \(ldouble b \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.65.3 Member Data Documentation

7.65.3.1 [ldouble SinPix::var](#) [[private](#)]

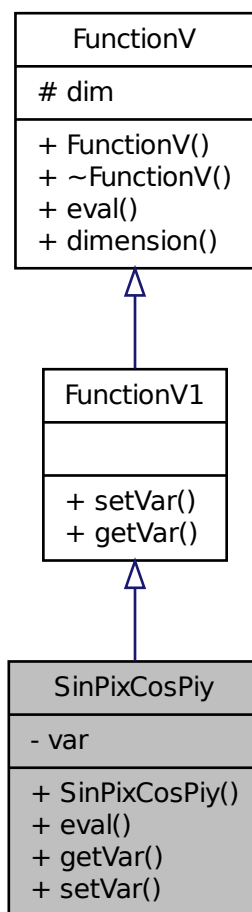
The documentation for this class was generated from the following file:

- [SinPix.hpp](#)

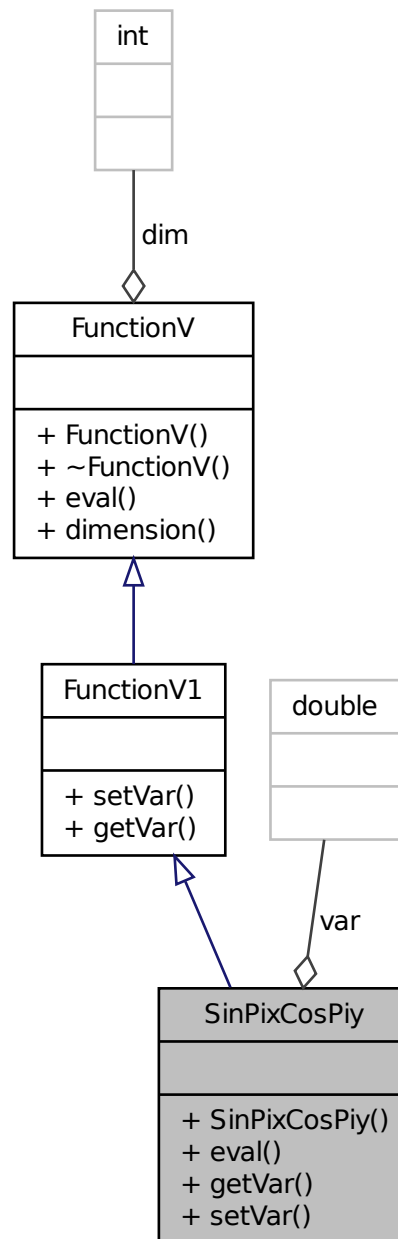
7.66 SinPixCosPiy Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.66.1 Constructor & Destructor Documentation

7.66.1.1 [SinPixCosPiy::SinPixCosPiy \(ldouble b \)](#) [[inline](#)]

7.66.2 Member Function Documentation

7.66.2.1 [ldouble SinPixCosPiy::eval \(int d, ldouble * x \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV](#).

7.66.2.2 [ldouble SinPixCosPiy::getVar \(void \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.66.2.3 [void SinPixCosPiy::setVar \(ldouble b \)](#) [[inline](#)],[[virtual](#)]

Implements [FunctionV1](#).

7.66.3 Member Data Documentation

7.66.3.1 [ldouble SinPixCosPiy::var](#) [[private](#)]

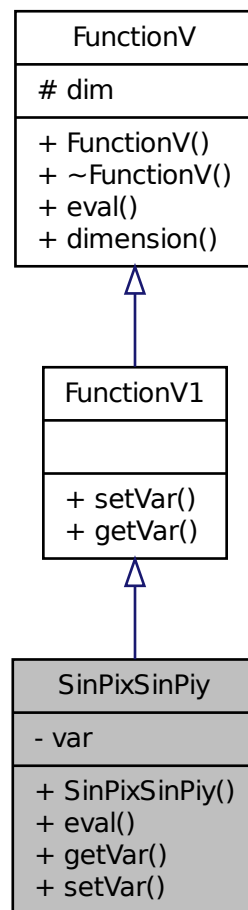
The documentation for this class was generated from the following file:

- [SinPixCosPiy.hpp](#)

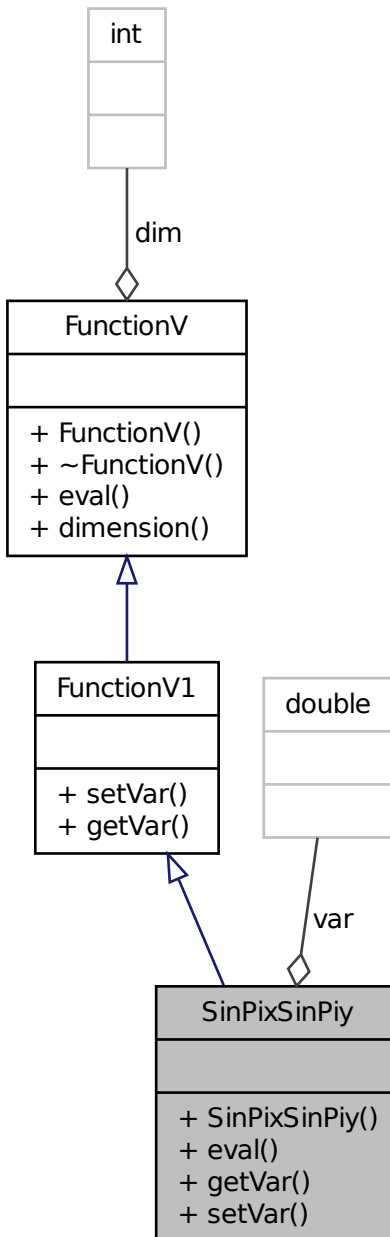
7.67 SinPixSinPiy Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.67.1 Constructor & Destructor Documentation

7.67.1.1 [SinPixSinPiy::SinPixSinPiy \(ldouble b \)](#) [inline]

7.67.2 Member Function Documentation

7.67.2.1 [ldouble SinPixSinPiy::eval \(int d, ldouble * x \)](#) [inline],[virtual]

Implements [FunctionV](#).

7.67.2.2 [ldouble SinPixSinPiy::getVar \(void \)](#) [inline],[virtual]

Implements [FunctionV1](#).

7.67.2.3 [void SinPixSinPiy::setVar \(ldouble b \)](#) [inline],[virtual]

Implements [FunctionV1](#).

7.67.3 Member Data Documentation

7.67.3.1 [ldouble SinPixSinPiy::var](#) [private]

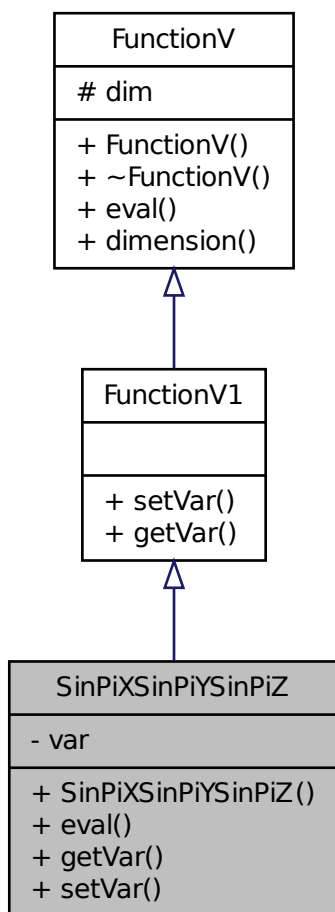
The documentation for this class was generated from the following file:

- [SinPixSinPiy.hpp](#)

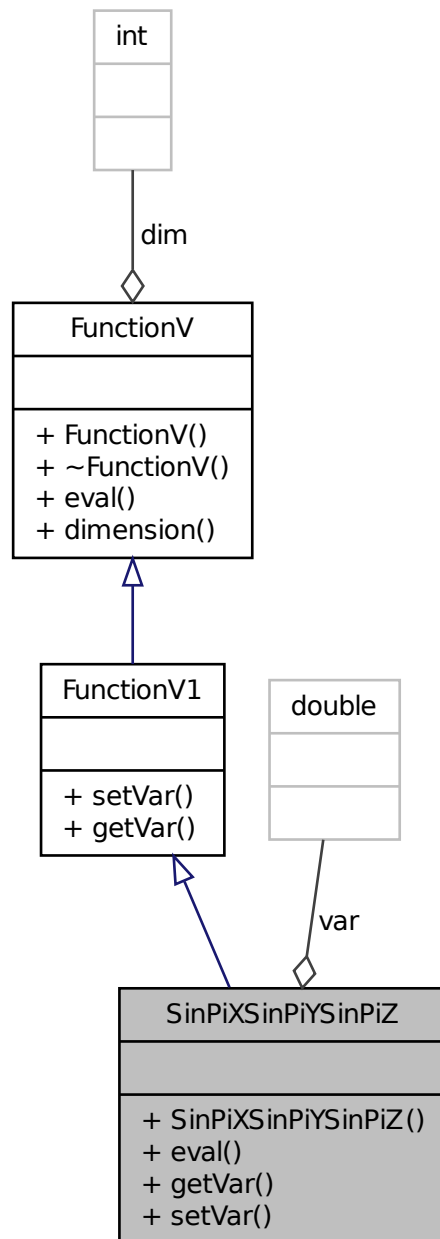
7.68 SinPiXSinPiYSinPiZ Class Reference

```
#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

- [ldouble var](#)

Additional Inherited Members

7.68.1 Constructor & Destructor Documentation

7.68.1.1 [SinPiXSinPiYSinPiZ::SinPiXSinPiYSinPiZ \(ldouble b \)](#) [[inline](#)]

7.68.2 Member Function Documentation

7.68.2.1 [ldouble SinPiXSinPiYSinPiZ::eval \(int d, ldouble * x \)](#) [[inline](#)], [[virtual](#)]

Implements [FunctionV](#).

7.68.2.2 [ldouble SinPiXSinPiYSinPiZ::getVar \(void \)](#) [[inline](#)], [[virtual](#)]

Implements [FunctionV1](#).

7.68.2.3 [void SinPiXSinPiYSinPiZ::setVar \(ldouble b \)](#) [[inline](#)], [[virtual](#)]

Implements [FunctionV1](#).

7.68.3 Member Data Documentation

7.68.3.1 [ldouble SinPiXSinPiYSinPiZ::var](#) [[private](#)]

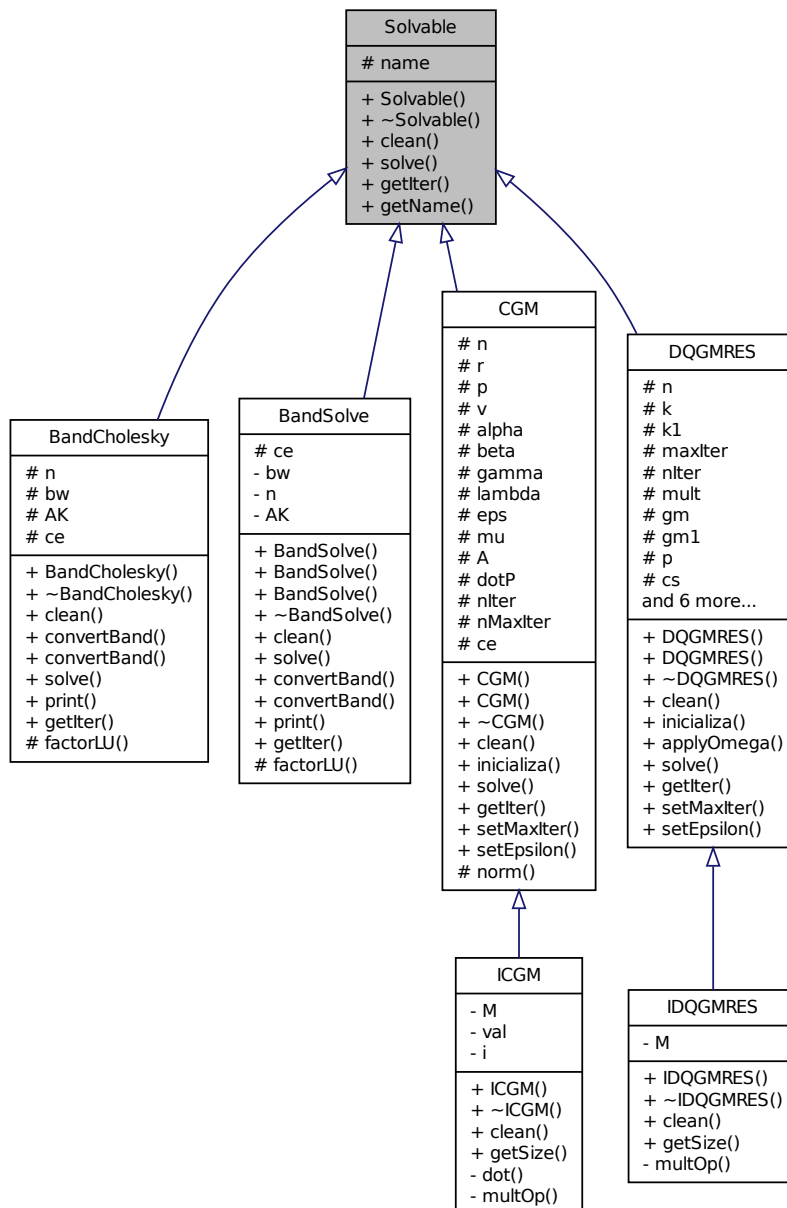
The documentation for this class was generated from the following file:

- [SinPiXSinPiYSinPiZ.hpp](#)

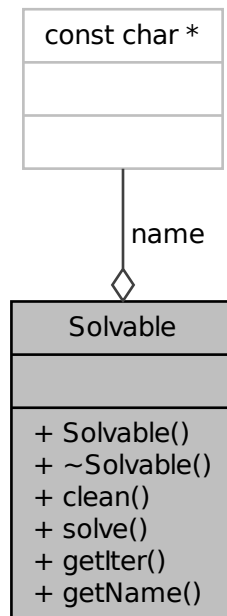
7.69 Solvable Class Reference

```
#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]`

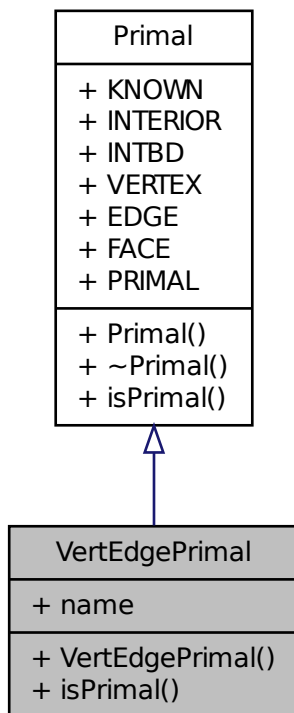
The documentation for this class was generated from the following file:

- [Solvable.hpp](#)

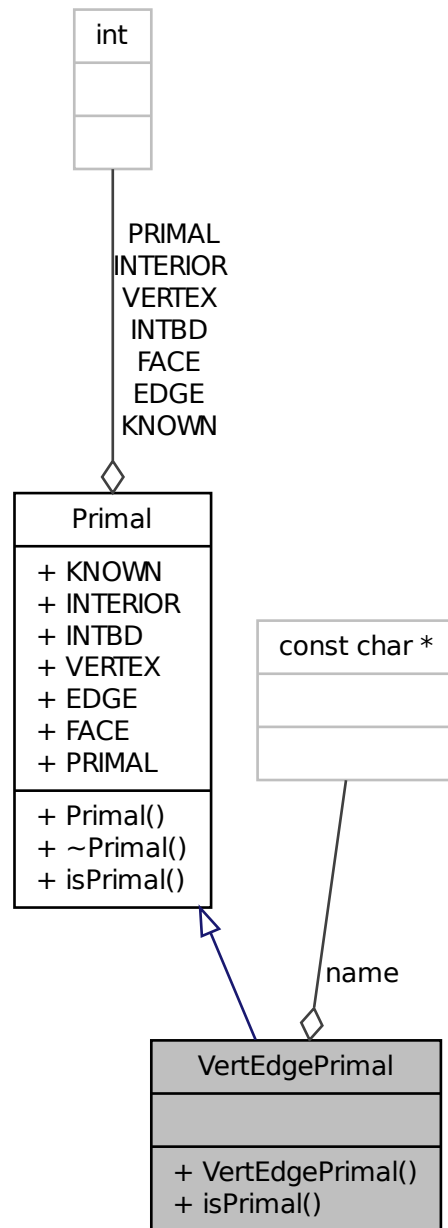
7.70 VertEdgePrimal Class Reference

```
#include <VertEdgePrimal.hpp>
```

Inheritance diagram for VertEdgePrimal:



Collaboration diagram for VertEdgePrimal:



Public Member Functions

- [VertEdgePrimal](#) (void)
- [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`

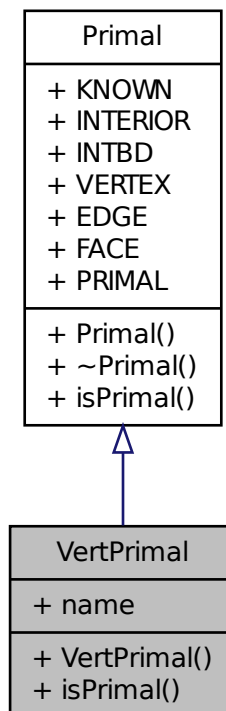
The documentation for this class was generated from the following file:

- [VertEdgePrimal.hpp](#)

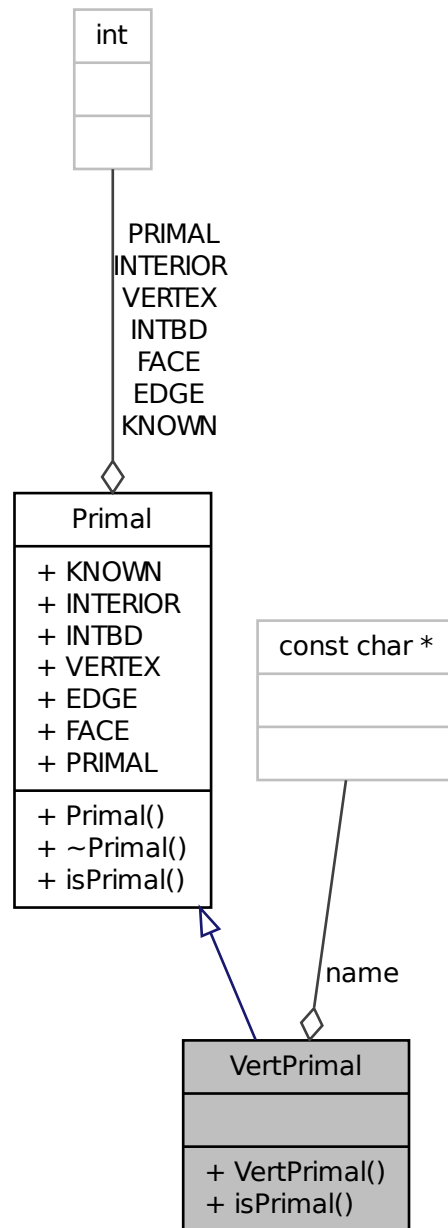
7.71 VertPrimal Class Reference

```
#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

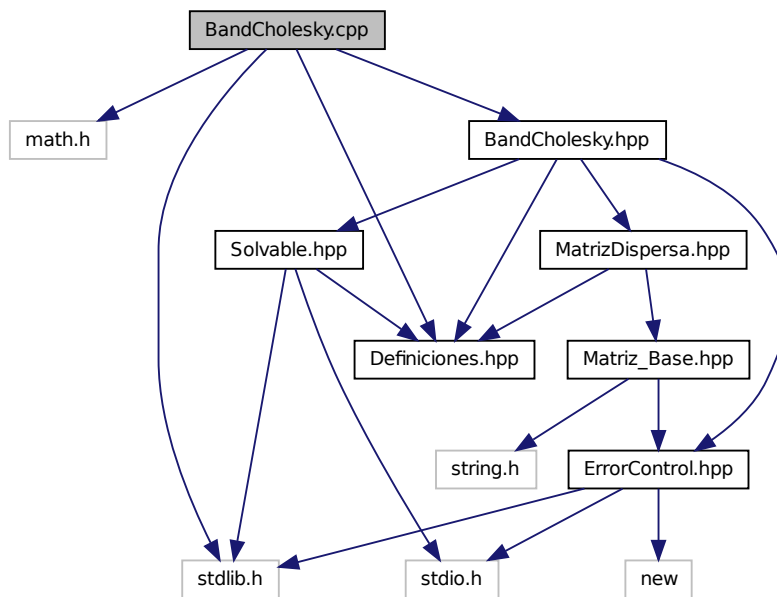
7.71.3.1 `const char* VertPrimal::name`

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

- [VertPrimal.hpp](#)

8.2 BandCholesky.cpp File Reference

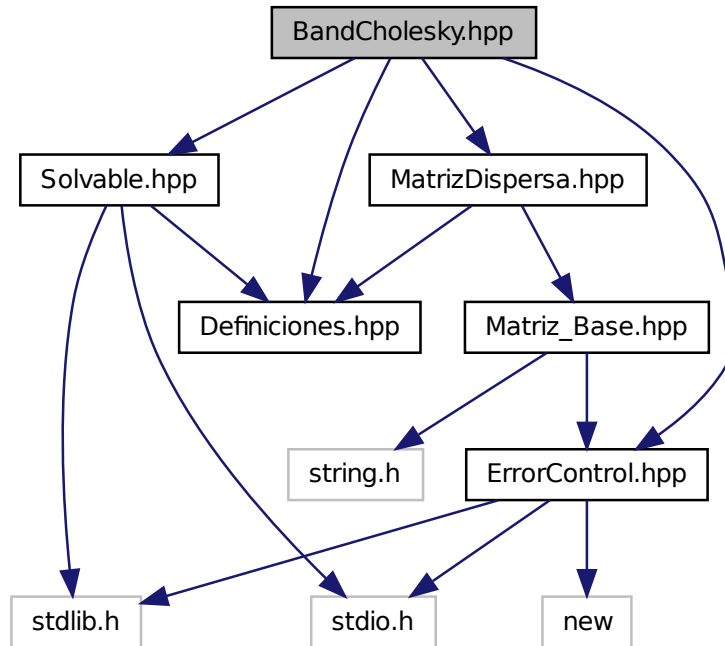
```
#include <math.h>
#include <stdlib.h>
#include "Definiciones.hpp"
#include "BandCholesky.hpp"
Include dependency graph for BandCholesky.cpp:
```



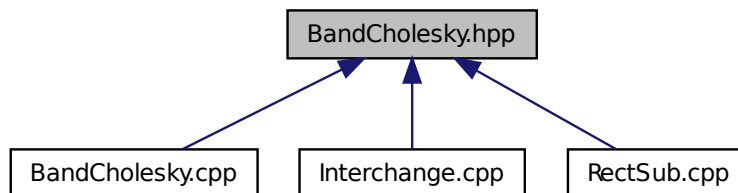
8.3 BandCholesky.hpp File Reference

```
#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:

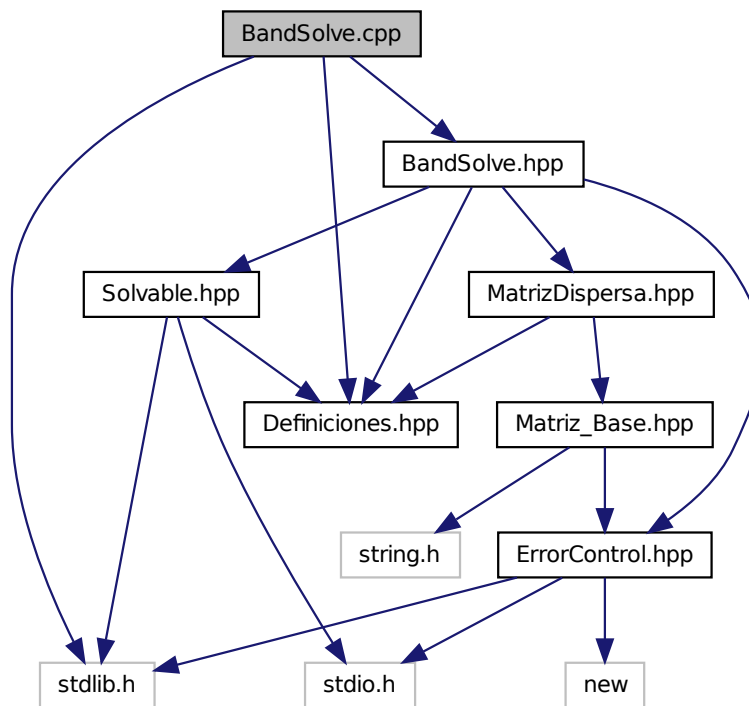


Classes

- class [BandCholesky](#)

8.4 BandSolve.cpp File Reference

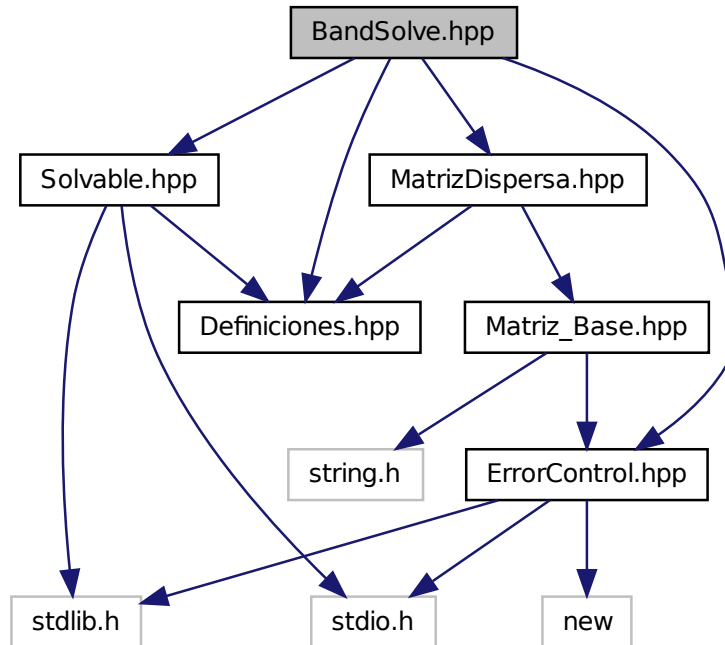
```
#include <stdlib.h>
#include "Definiciones.hpp"
#include "BandSolve.hpp"
Include dependency graph for BandSolve.cpp:
```



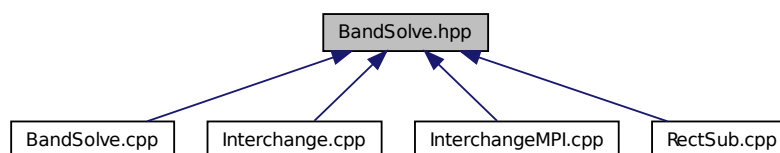
8.5 BandSolve.hpp File Reference

```
#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:

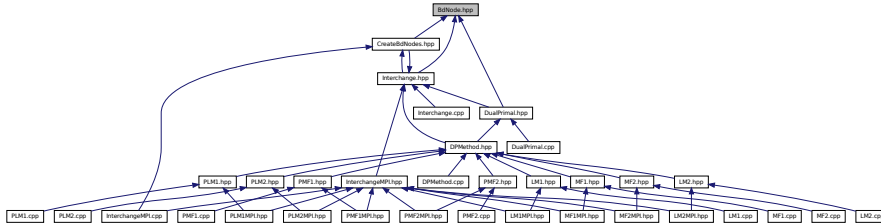


Classes

- class [BandSolve](#)

8.6 BdNode.hpp File Reference

This graph shows which files directly or indirectly include this file:

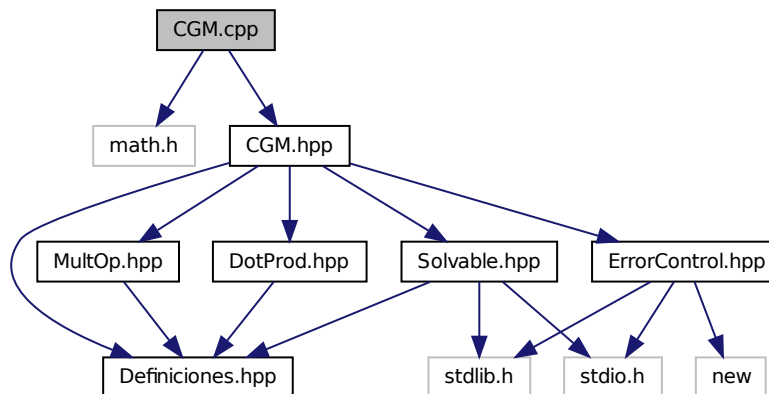


Classes

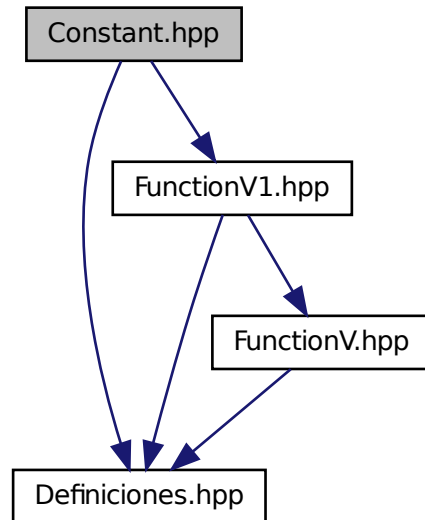
- class [BdNode](#)

8.7 CGM.cpp File Reference

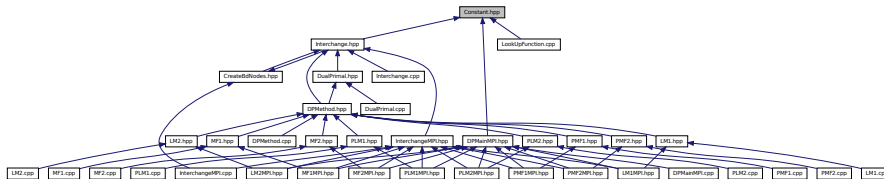
```
#include <math.h>
#include "CGM.hpp"
Include dependency graph for CGM.cpp:
```



Include dependency graph for Constant.hpp:



This graph shows which files directly or indirectly include this file:



Classes

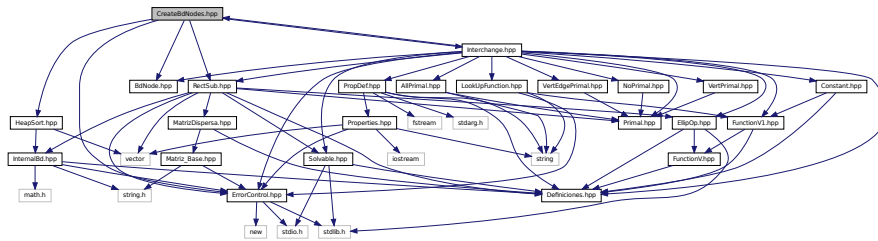
- class [Constant](#)

8.10 CreateBdNodes.hpp File Reference

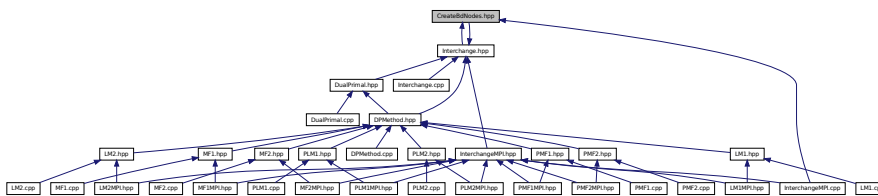
```

#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:

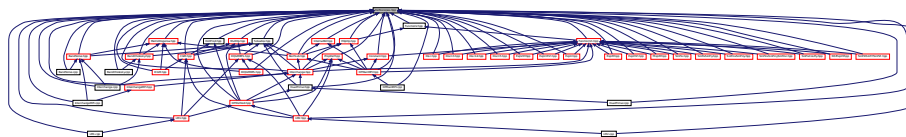


Classes

- class [CreateBdNodes](#)

8.11 Definiciones.hpp File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [NMAXITER](#) 200
Numero maximo de iteraiones en los metodos iterativos.
- #define [NMAXITER_LOCAL](#) 50000
- #define [EPSILON](#) 1e-6
Tolerancia en los metodos iterativos.
- #define [EPSILON_LOCAL](#) (EPSILON/1e+2)
- #define [EPS_EQUAL](#) 1e-15
Se toman como iguales dos nodos que difieran en menos que esta EPS_EQUAL.

- `#define RESIDUAL`
Con esta opcion visualiza o no el residual de cada iteracion.
- `#define DIM_VECTOR 1`
Dimension del vector (1) escalar.
- `#define COEFICIENTES_CONSTANTES`
Con esta opcion se calcula el numero de condicionamiento en los metodos preconditionados.
- `#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 preconditionados.

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`

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+2)`

8.11.1.7 `#define NMAXITER 200`

Numero maximo de iteraiones en los metodos iterativos.

8.11.1.8 `#define NMAXITER_LOCAL 50000`

8.11.1.9 `#define RESIDUAL`

Con esta opcion visualiza o no el residual de cada iteracion.

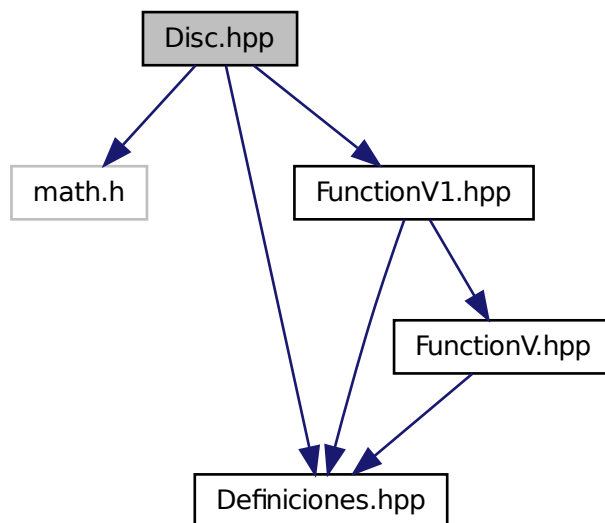
8.11.2 Typedef Documentation

8.11.2.1 `typedef double ldouble`

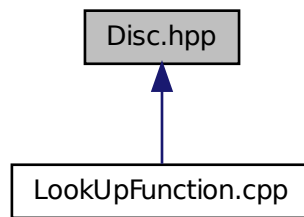
Define ldouble como double.

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:



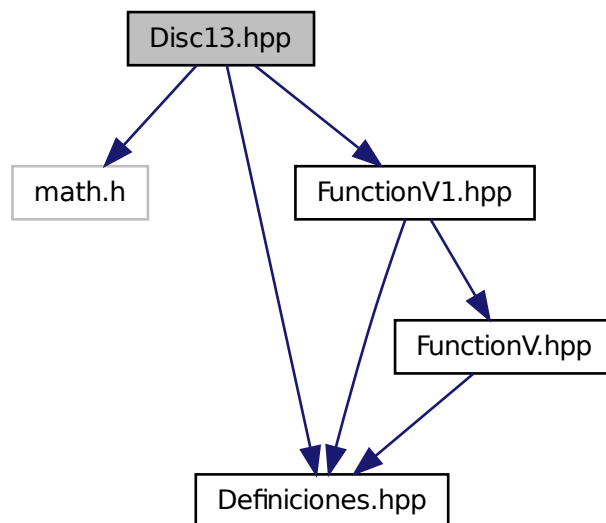
Classes

- class [Disc](#)

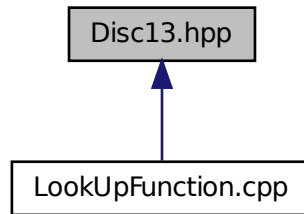
8.13 Disc13.hpp File Reference

```
#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:



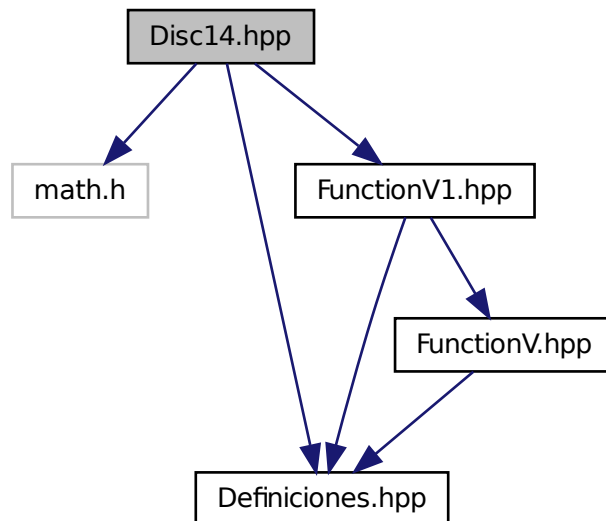
Classes

- class [Disc13](#)

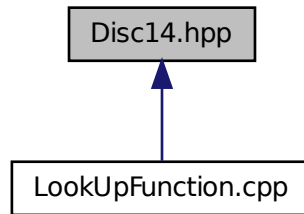
8.14 Disc14.hpp File Reference

```
#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:



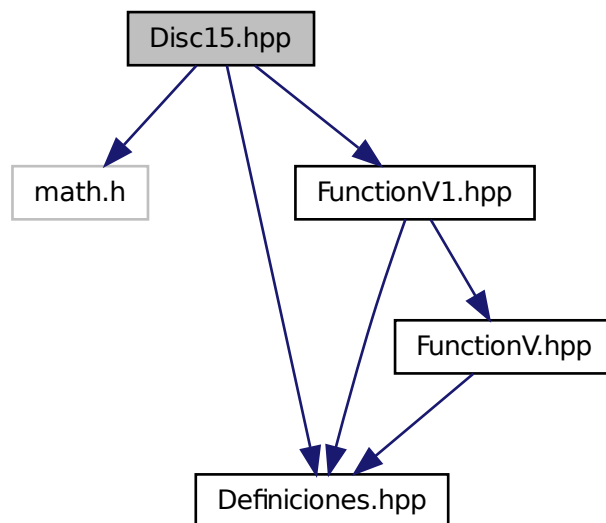
Classes

- class [Disc14](#)

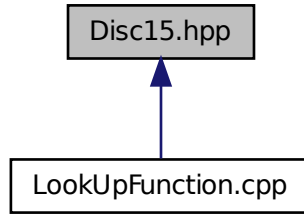
8.15 Disc15.hpp File Reference

```
#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:

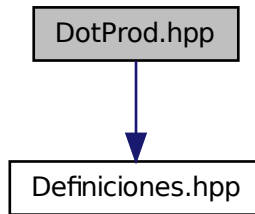


Classes

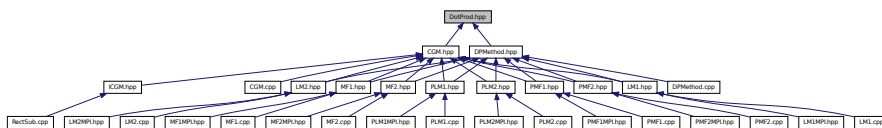
- class [Disc15](#)

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:

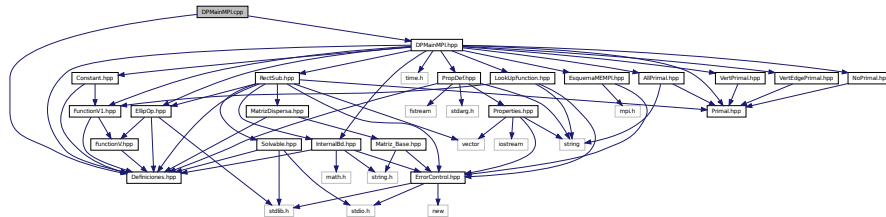


Classes

- class [DotProd](#)

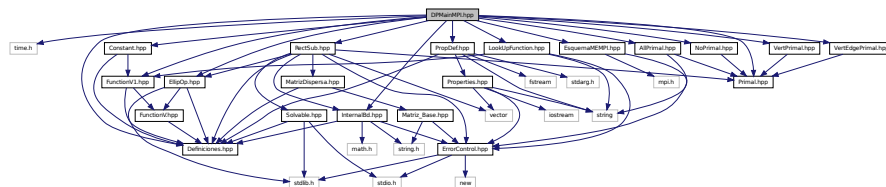
8.17 DPMainMPI.cpp File Reference

```
#include "Definiciones.hpp"
#include "DPMainMPI.hpp"
Include dependency graph for DPMainMPI.cpp:
```



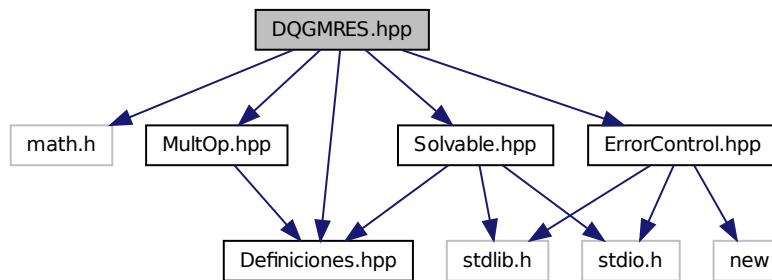
8.18 DPMainMPI.hpp File Reference

```
#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:
```

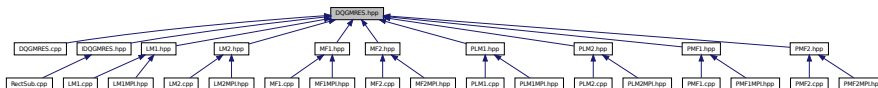


8.22 DQGMRES.hpp File Reference

```
#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:



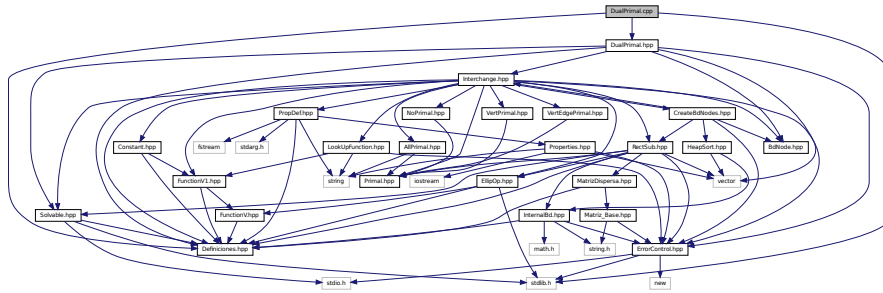
Classes

- class [DQGMRES](#)

8.23 DualPrimal.cpp File Reference

```
#include <stdlib.h>
#include "Definiciones.hpp"
#include "DualPrimal.hpp"
```

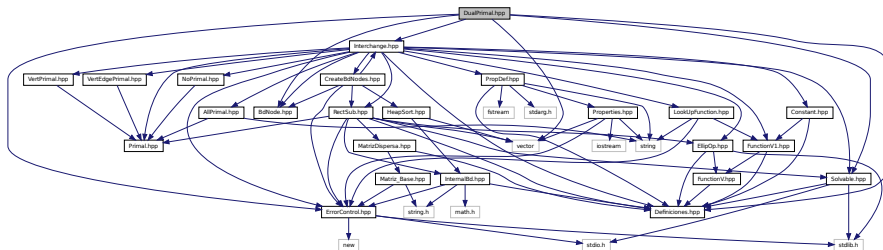
Include dependency graph for DualPrimal.cpp:



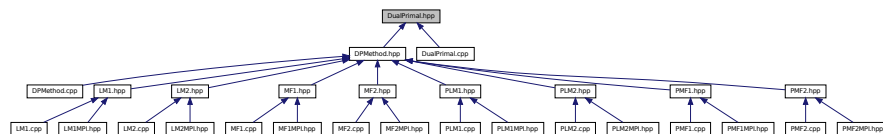
8.24 DualPrimal.hpp File Reference

```
#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:

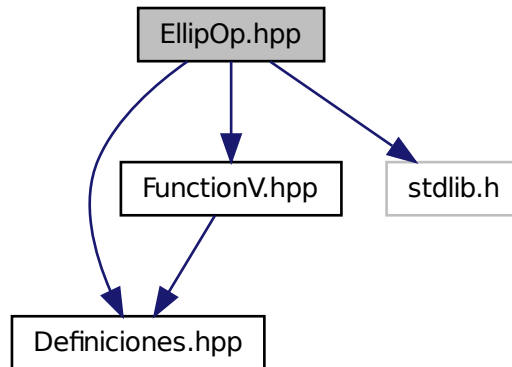


Classes

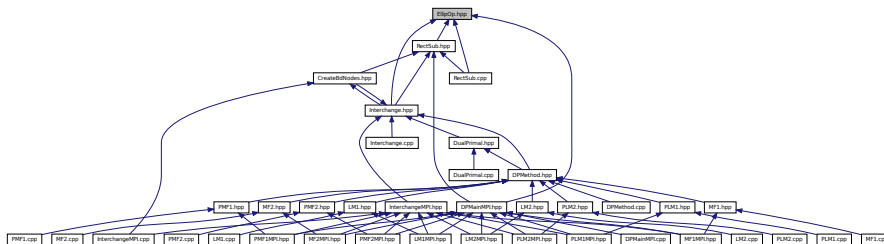
- class [DualPrimal](#)

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:



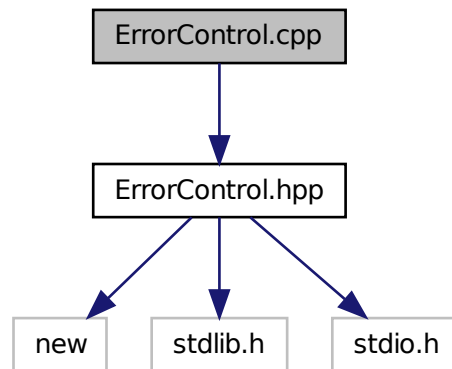
Classes

- class [EllipOp](#)

8.26 ErrorControl.cpp File Reference

```
#include "ErrorControl.hpp"
```

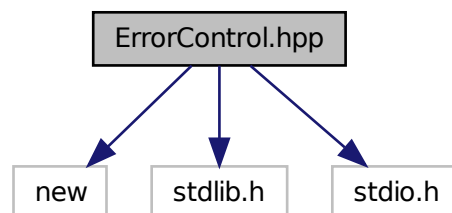
Include dependency graph for ErrorControl.cpp:



8.27 ErrorControl.hpp File Reference

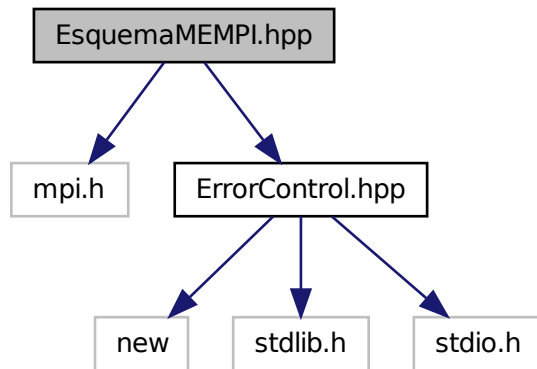
```
#include <new>  
#include <stdlib.h>  
#include <stdio.h>
```

Include dependency graph for ErrorControl.hpp:

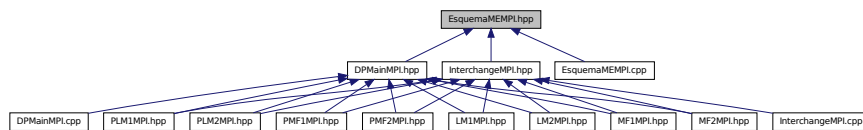


8.29 EsquemaMEMPI.hpp File Reference

```
#include "mpi.h"
#include "ErrorControl.hpp"
Include dependency graph for EsquemaMEMPI.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

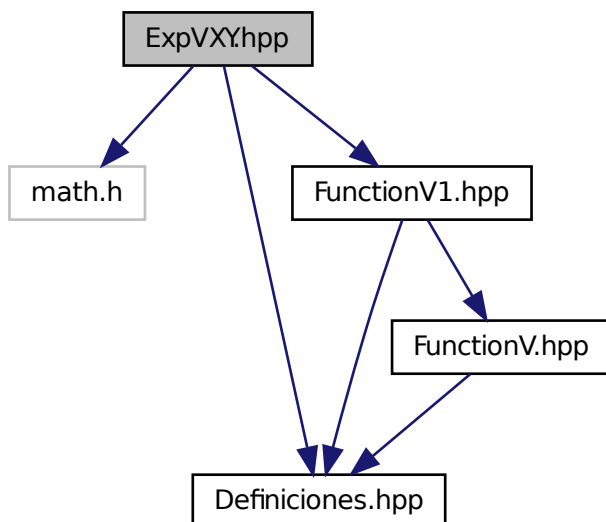
- class [EsquemaMEMPI](#)

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

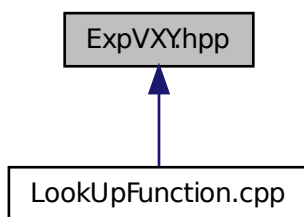
8.30 ExpVXY.hpp File Reference

```
#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:



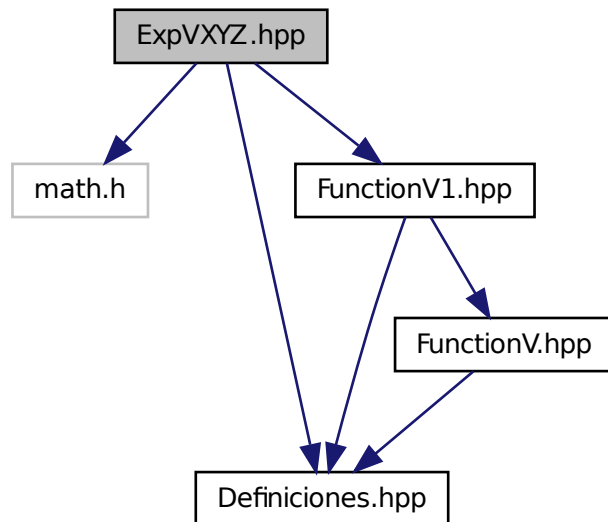
Classes

- class [ExpVXY](#)

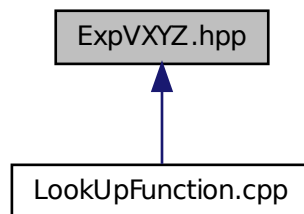
8.31 ExpVXYZ.hpp File Reference

```
#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:

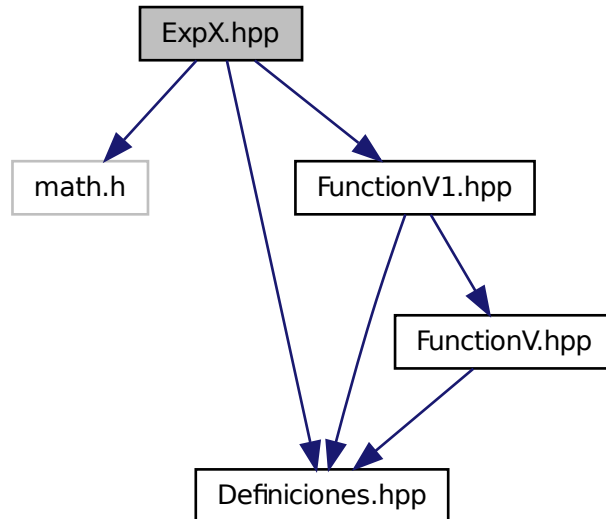


Classes

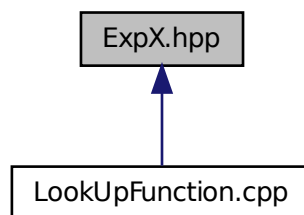
- class [ExpVXYZ](#)

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:

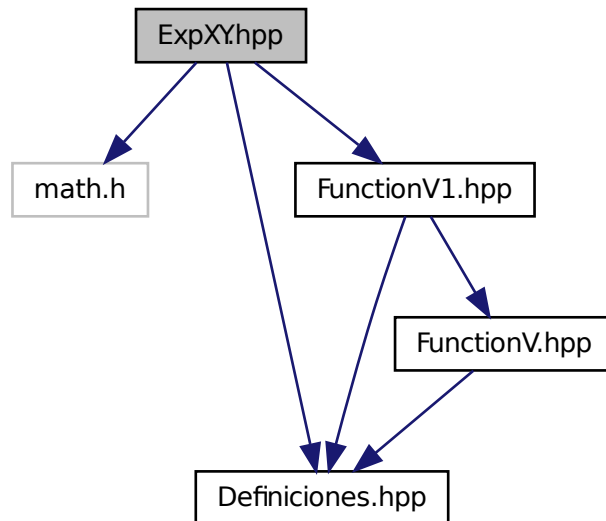


Classes

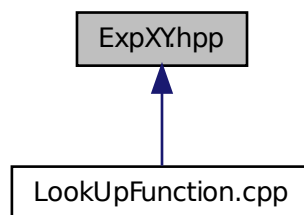
- class [ExpX](#)

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:

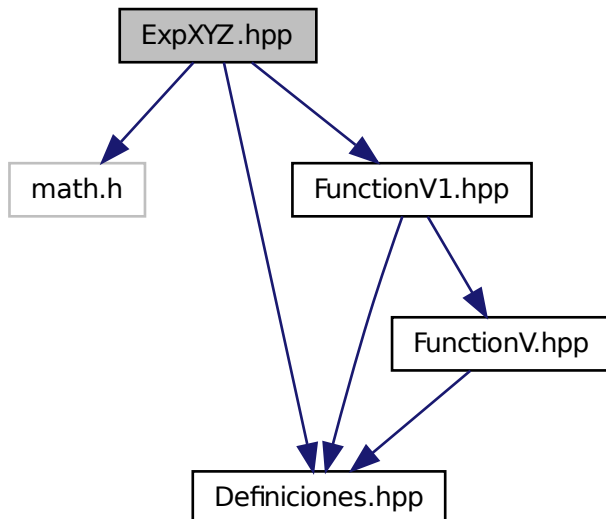


Classes

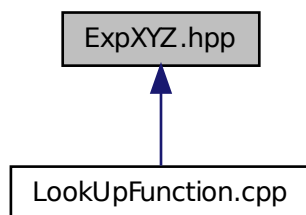
- class [ExpXY](#)

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:

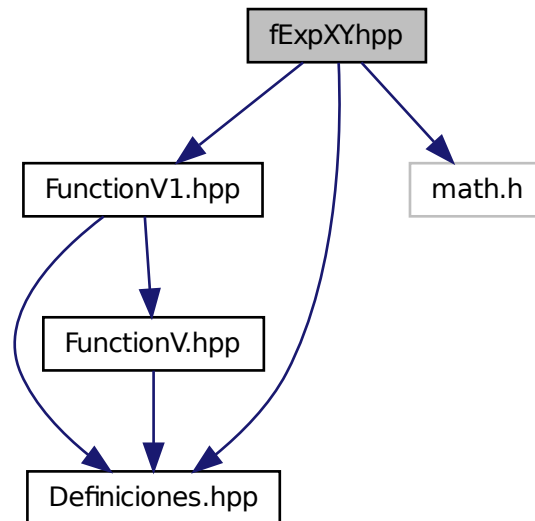


Classes

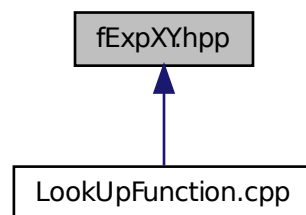
- class [ExpXYZ](#)

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:



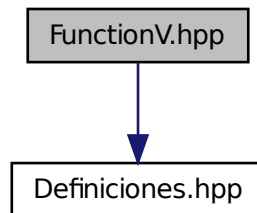
Classes

- class [fExpXY](#)

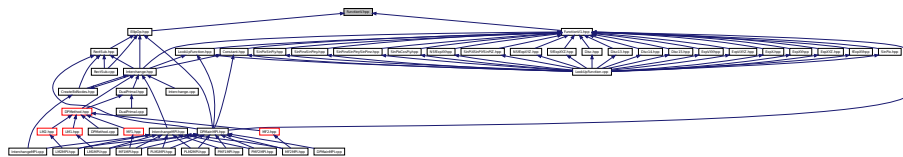
8.36 FunctionV.hpp File Reference

```
#include "Definiciones.hpp"
```

Include dependency graph for FunctionV.hpp:



This graph shows which files directly or indirectly include this file:



Classes

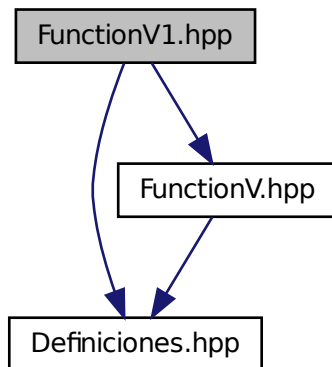
- class [FunctionV](#)

8.37 FunctionV1.hpp File Reference

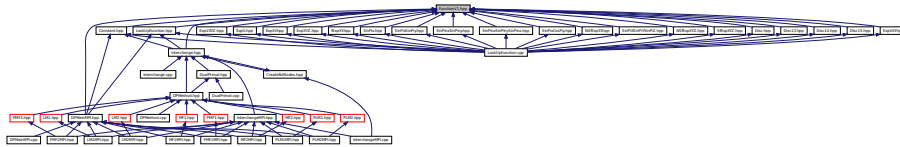
```
#include "Definiciones.hpp"
```

```
#include "FunctionV.hpp"
```

Include dependency graph for FunctionV1.hpp:



This graph shows which files directly or indirectly include this file:



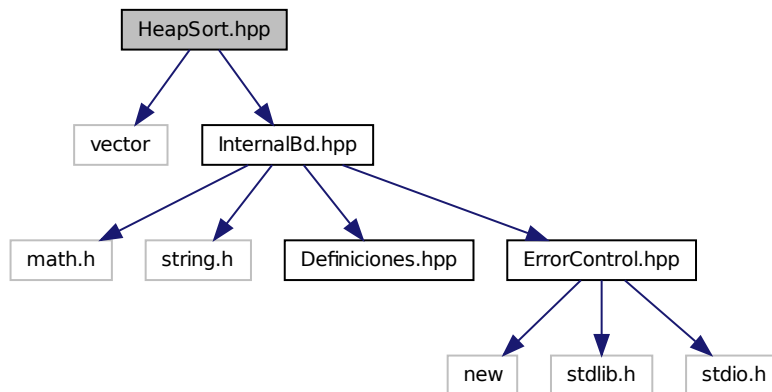
Classes

- class [FunctionV1](#)

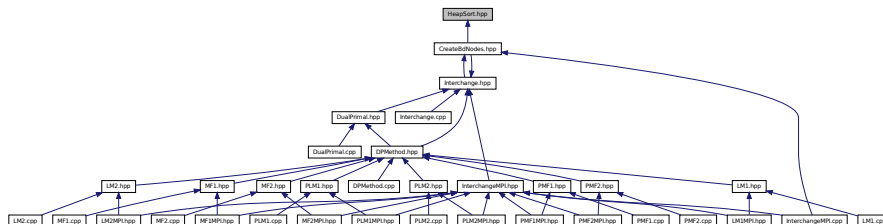
8.38 HeapSort.hpp File Reference

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

Include dependency graph for HeapSort.hpp:



This graph shows which files directly or indirectly include this file:



Classes

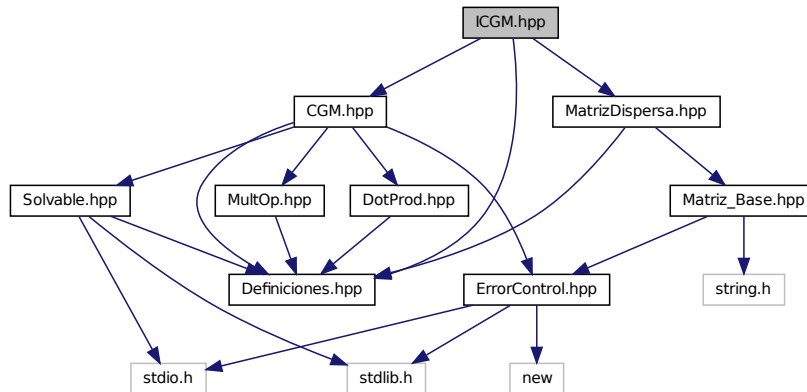
- class [HeapSort](#)

8.39 ICGM.hpp File Reference

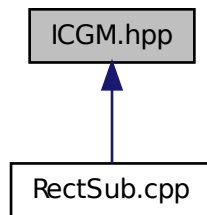
```

#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.

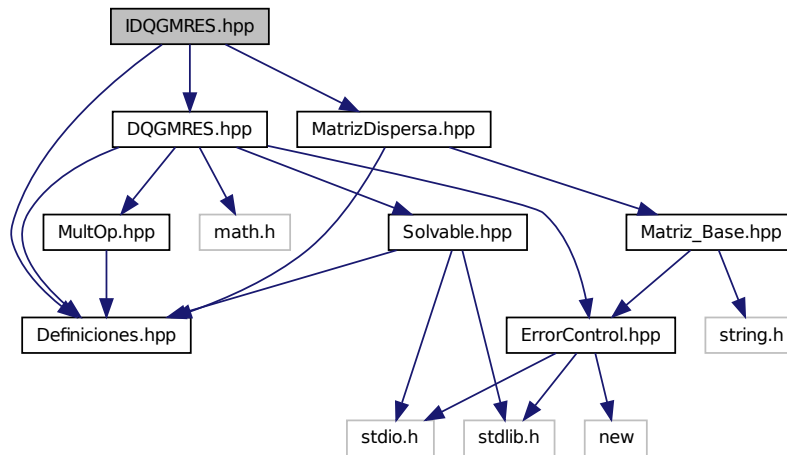
8.40 IDQGMRES.hpp File Reference

```

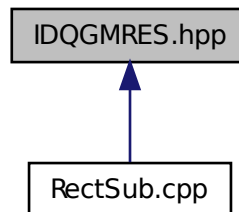
#include "Definiciones.hpp"
#include "DQGMRES.hpp"
#include "MatrizDispersa.hpp"

```

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.

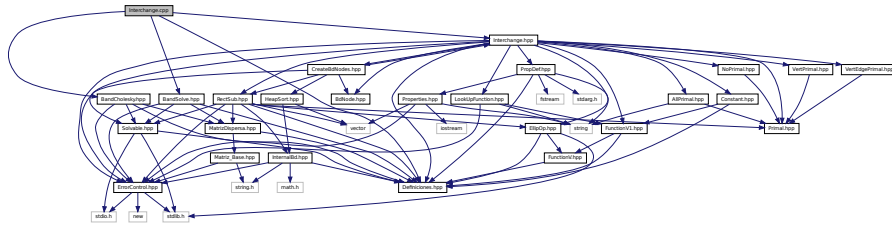
8.41 Interchange.cpp File Reference

```

#include "Definiciones.hpp"
#include "Interchange.hpp"
#include "BandSolve.hpp"
#include "BandCholesky.hpp"

```

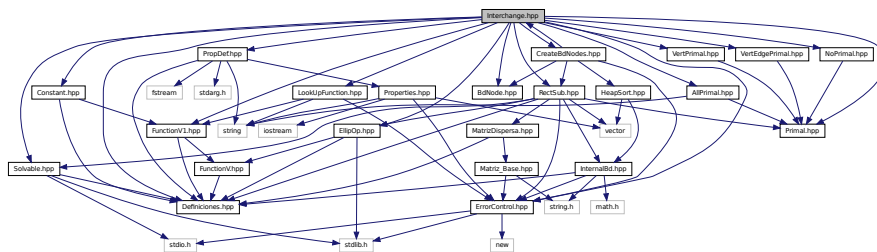
Include dependency graph for Interchange.cpp:



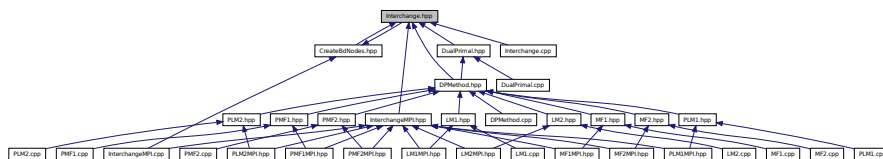
8.42 Interchange.hpp File Reference

```
#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:

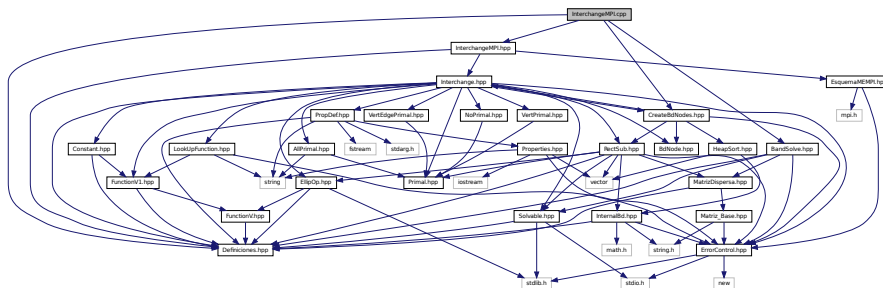


Classes

- class [Interchange](#)

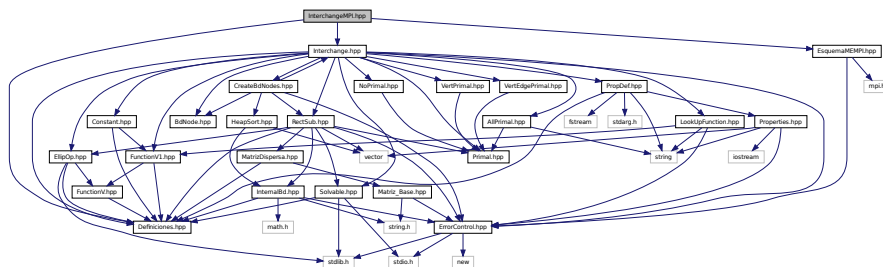
8.43 InterchangeMPI.cpp File Reference

```
#include "Definiciones.hpp"
#include "InterchangeMPI.hpp"
#include "BandSolve.hpp"
#include "CreateBdNodes.hpp"
Include dependency graph for InterchangeMPI.cpp:
```

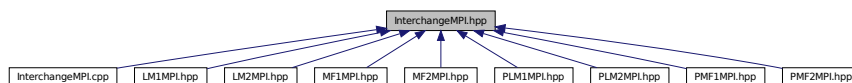


8.44 InterchangeMPI.hpp File Reference

```
#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:

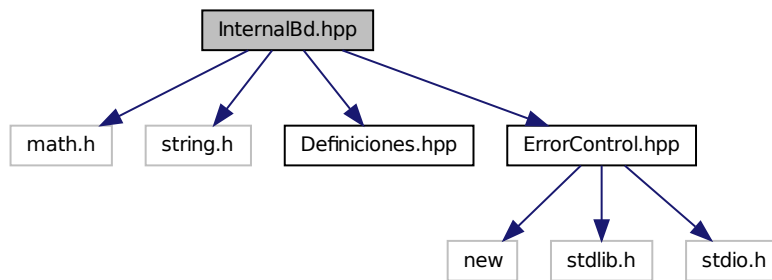


Classes

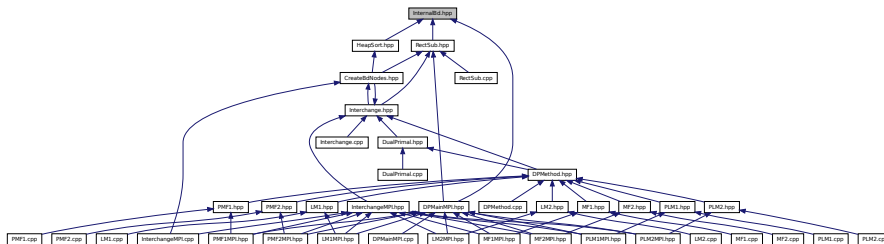
- class [InterchangeMPI](#)

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:



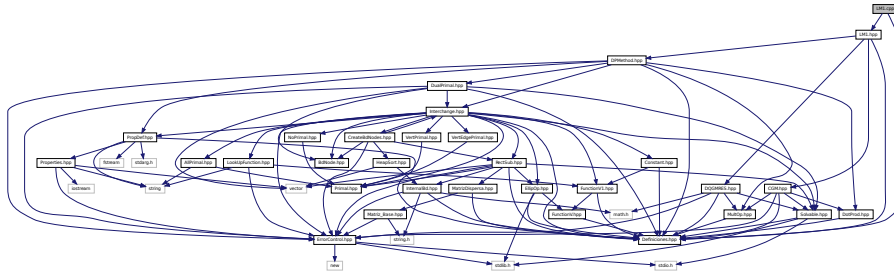
Classes

- class [InternalBd](#)

8.46 LM1.cpp File Reference

```
#include "Definiciones.hpp"
#include "LM1.hpp"
```

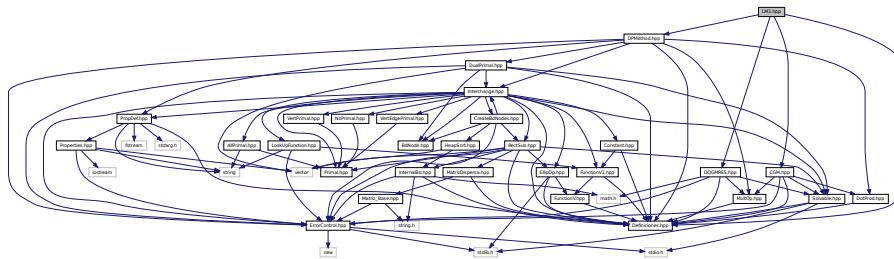
Include dependency graph for LM1.cpp:



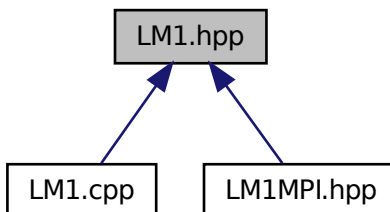
8.47 LM1.hpp File Reference

```
#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:

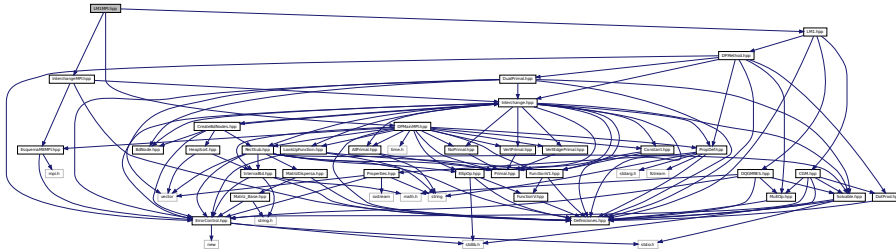


Classes

- class [LM1](#)

8.48 LM1MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "LM1.hpp"
Include dependency graph for LM1MPI.hpp:
```

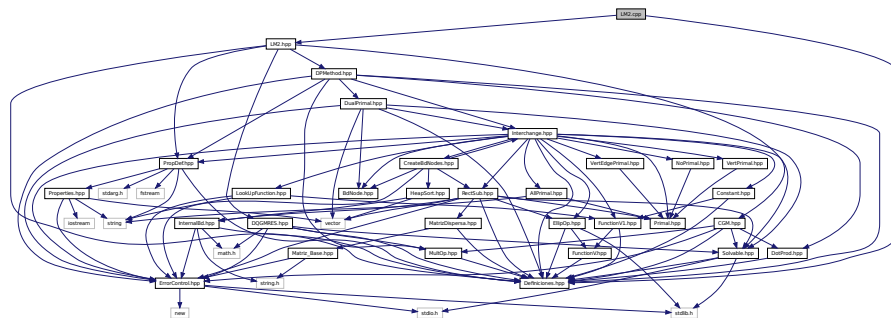


Classes

- class [LM1MPI](#)
Clase para definir el metodo LM-1 de DVS-DDM.

8.49 LM2.cpp File Reference

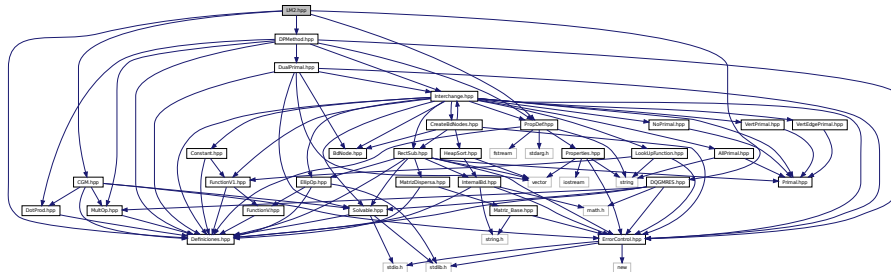
```
#include "Definiciones.hpp"
#include "LM2.hpp"
Include dependency graph for LM2.cpp:
```



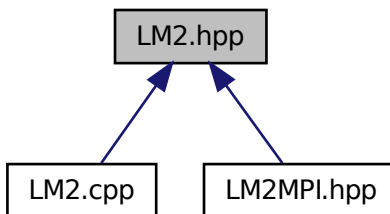
8.50 LM2.hpp File Reference

```
#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:



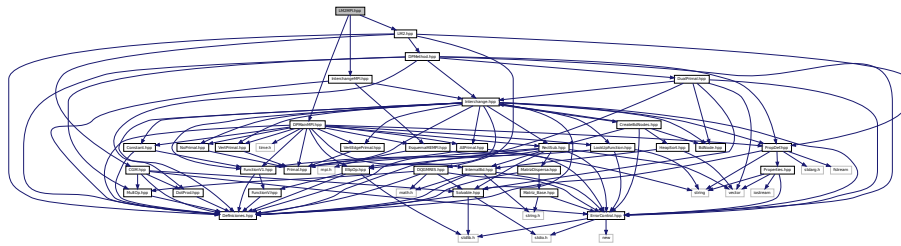
Classes

- 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.

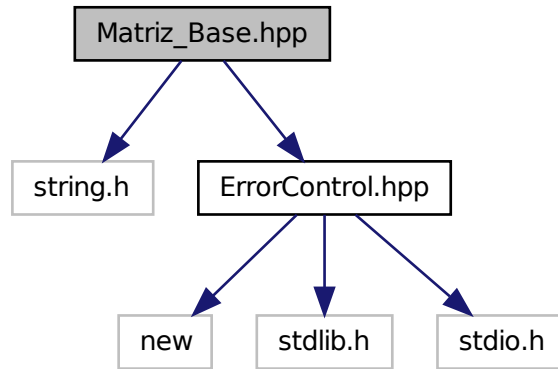
8.52 LookUpFunction.cpp File Reference

```
#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>
```

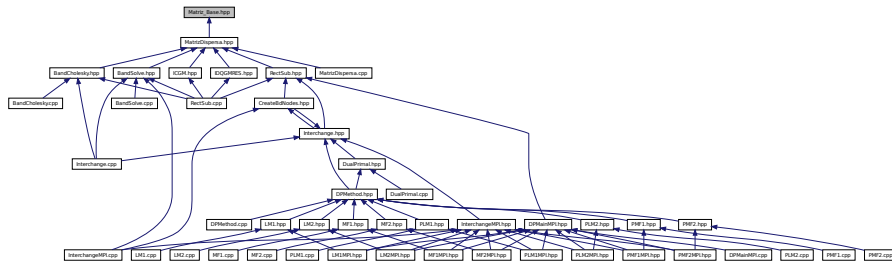
Include dependency graph for LookUpFunction.cpp:



Include dependency graph for Matriz_Base.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [Matriz_Base](#)

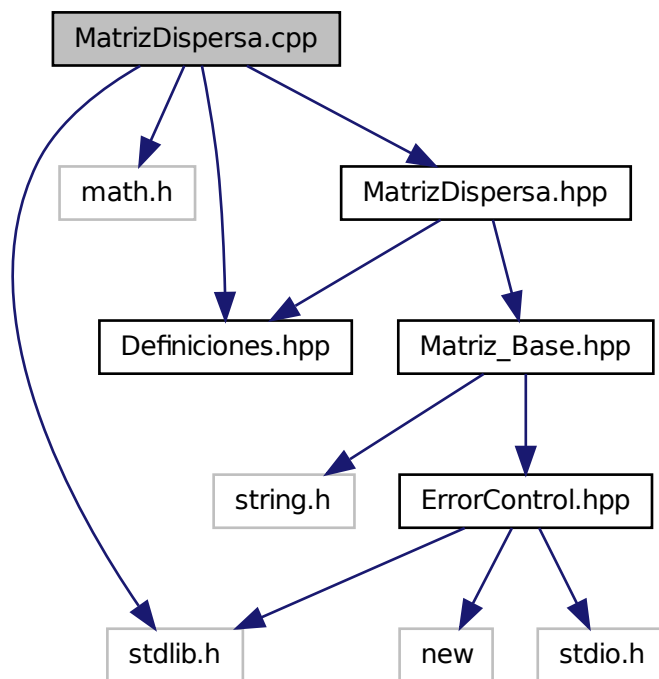
Clase base para el trabajar con matrices.

8.56 MatrizDispersa.cpp File Reference

```

#include <stdlib.h>
#include <math.h>
#include "Definiciones.hpp"
#include "MatrizDispersa.hpp"
  
```

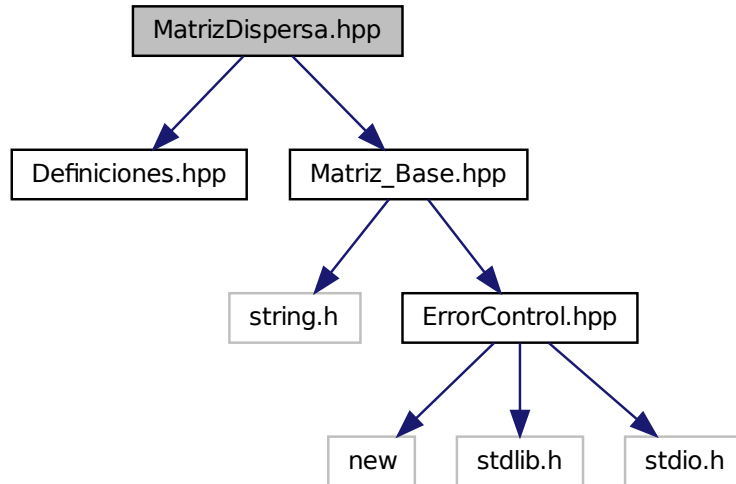

Include dependency graph for MatrizDispersa.cpp:



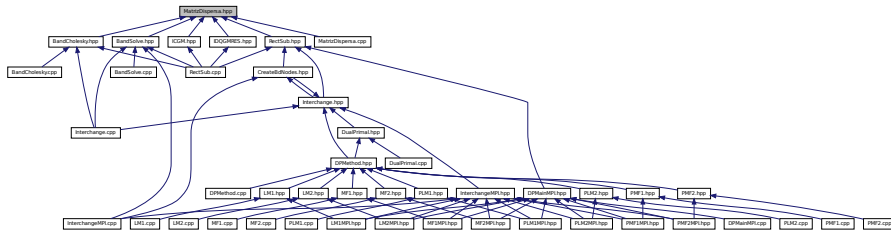
8.57 MatrizDispersa.hpp File Reference

```
#include "Definiciones.hpp"  
#include "Matriz_Base.hpp"
```

Include dependency graph for MatrizDispersa.hpp:



This graph shows which files directly or indirectly include this file:



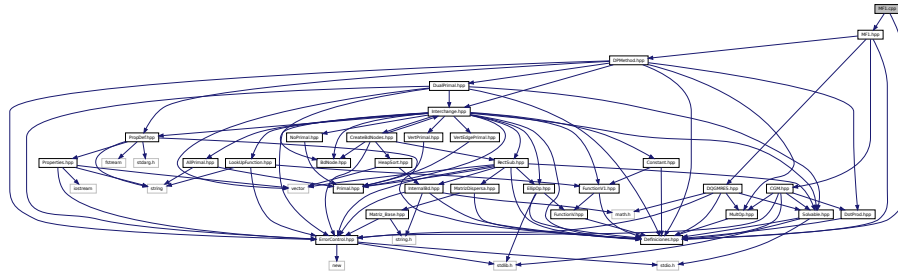
Classes

- class [MatrizDispersa](#)

8.58 MF1.cpp File Reference

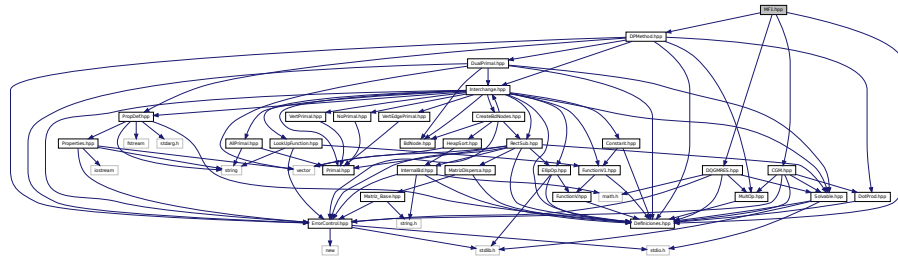
```
#include "Definiciones.hpp"
#include "MF1.hpp"
```

Include dependency graph for MF1.cpp:

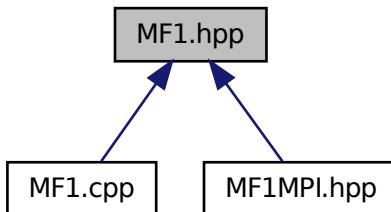


8.59 MF1.hpp File Reference

```
#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:

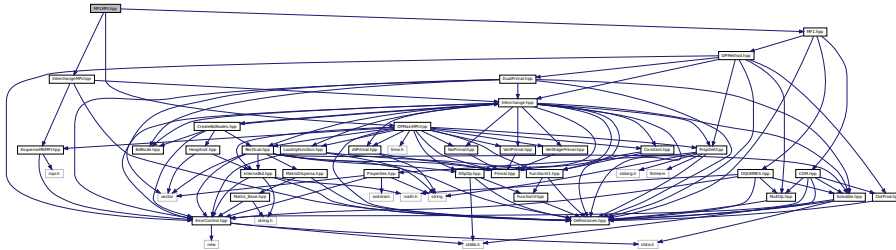


Classes

- class [MF1](#)

8.60 MF1MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "MF1.hpp"
Include dependency graph for MF1MPI.hpp:
```

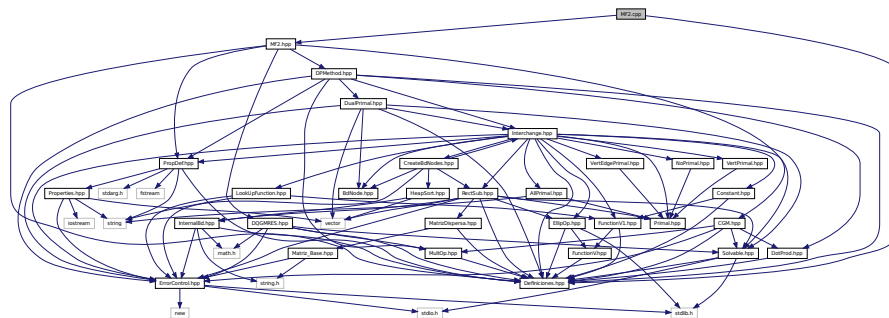


Classes

- class [MF1MPI](#)
Clase para definir el metodo MF-1 de DVS-DDM.

8.61 MF2.cpp File Reference

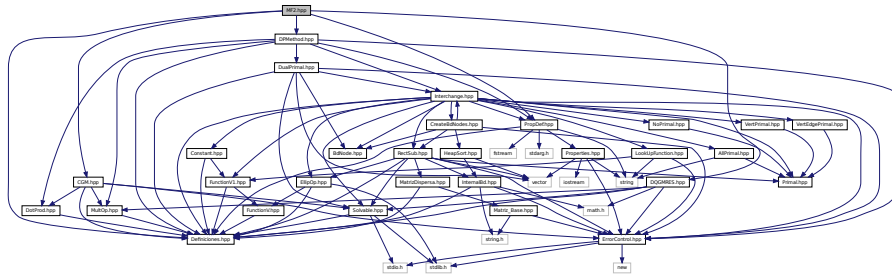
```
#include "Definiciones.hpp"
#include "MF2.hpp"
Include dependency graph for MF2.cpp:
```



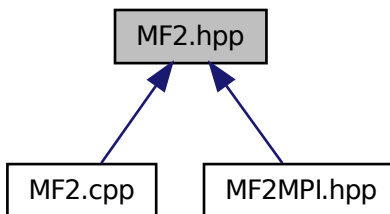
8.62 MF2.hpp File Reference

```
#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:



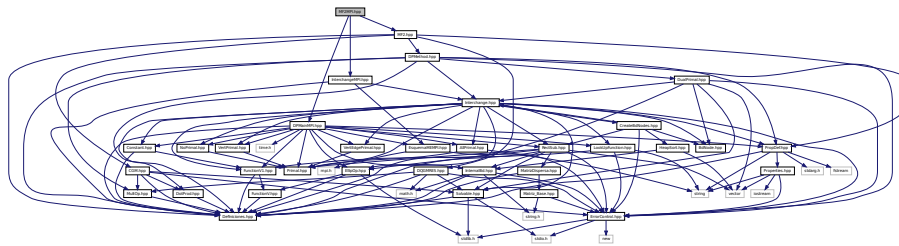
Classes

- class [MF2](#)

8.63 MF2MPI.hpp File Reference

```
#include "DPMMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "MF2.hpp"
```

Include dependency graph for MF2MPI.hpp:



Classes

- class [MF2MPI](#)

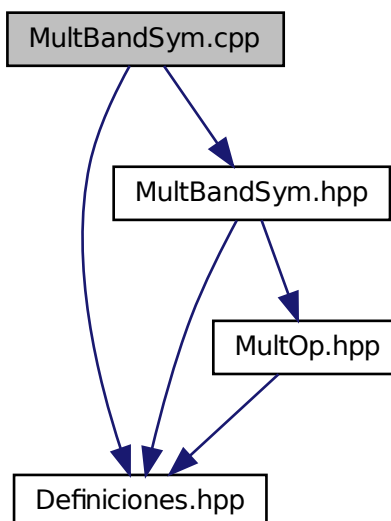
Clase para definir el metodo MF-2 de DVS-DDM.

8.64 MultBandSym.cpp File Reference

```
#include "Definiciones.hpp"
```

```
#include "MultBandSym.hpp"
```

Include dependency graph for MultBandSym.cpp:

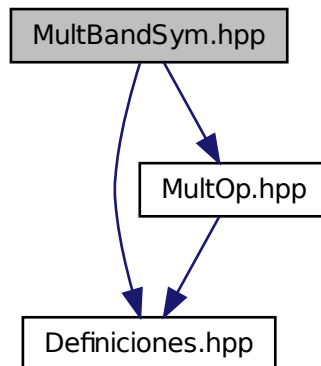


8.65 MultBandSym.hpp File Reference

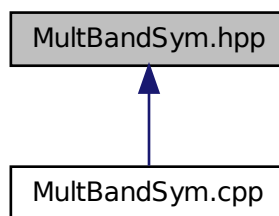
```
#include "Definiciones.hpp"
```

```
#include "MultOp.hpp"
```

Include dependency graph for MultBandSym.hpp:



This graph shows which files directly or indirectly include this file:



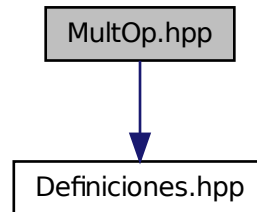
Classes

- class [MultBandSym](#)

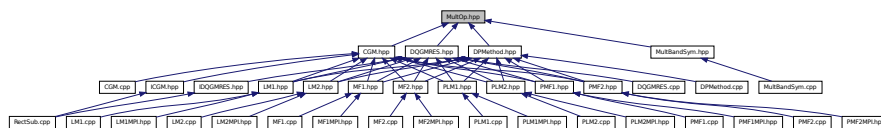
8.66 MultOp.hpp File Reference

```
#include "Definiciones.hpp"
```

Include dependency graph for MultOp.hpp:



This graph shows which files directly or indirectly include this file:



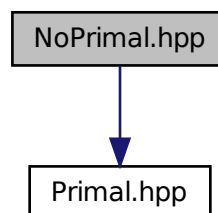
Classes

- class [MultOp](#)

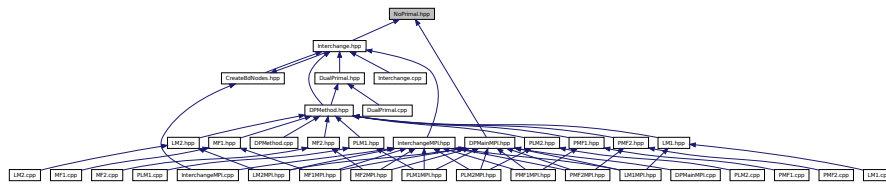
8.67 NoPrimal.hpp File Reference

```
#include "Primal.hpp"
```

Include dependency graph for NoPrimal.hpp:



This graph shows which files directly or indirectly include this file:

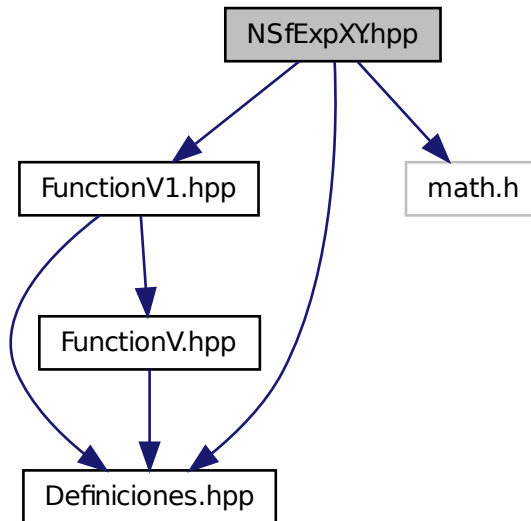


Classes

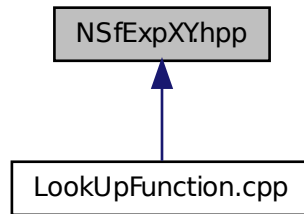
- class [NoPrimal](#)

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:

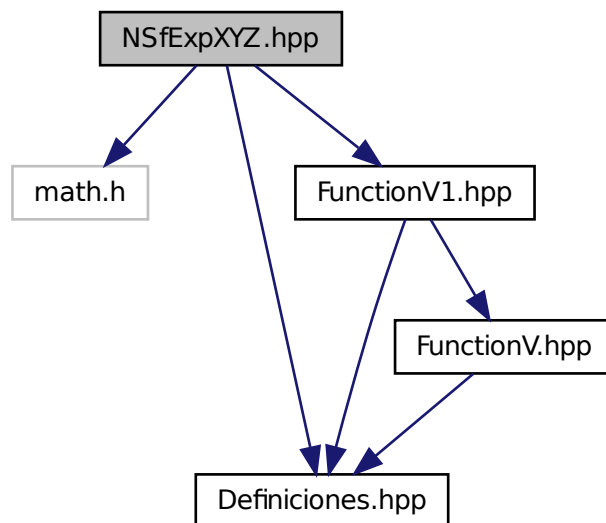


Classes

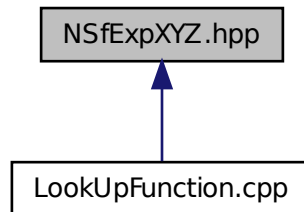
- class [NSfExpXY](#)

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:

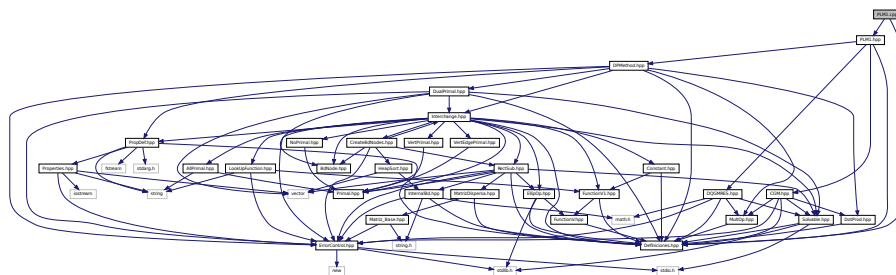


Classes

- class [NSfExpXYZ](#)

8.70 PLM1.cpp File Reference

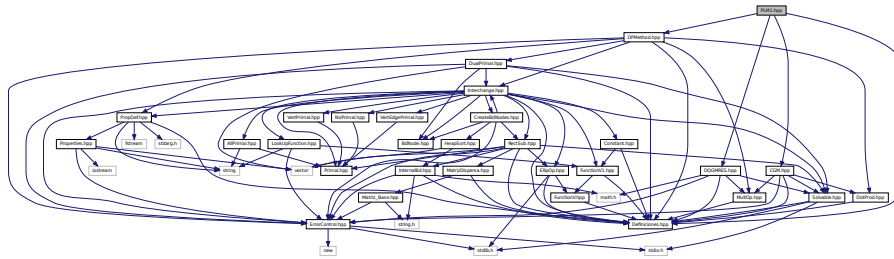
```
#include "Definiciones.hpp"
#include "PLM1.hpp"
Include dependency graph for PLM1.cpp:
```



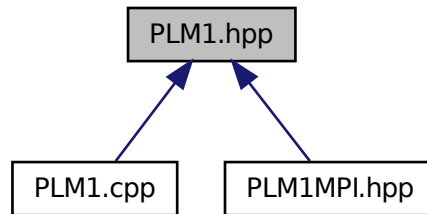
8.71 PLM1.hpp File Reference

```
#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:



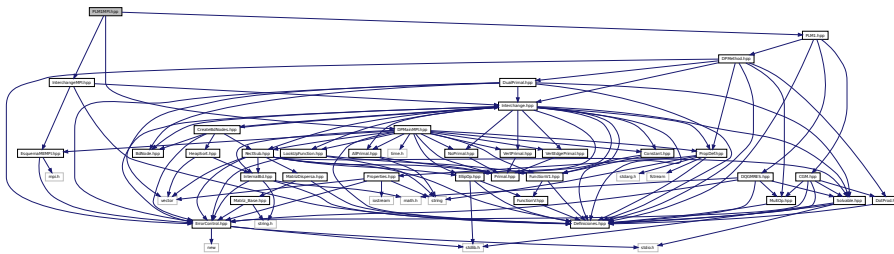
Classes

- class [PLM1](#)

8.72 PLM1MPI.hpp File Reference

```
#include "DPMMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PLM1.hpp"
```

Include dependency graph for PLM1MPI.hpp:



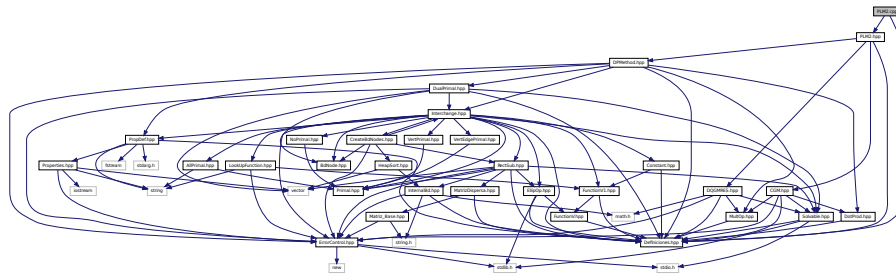
Classes

- class [PLM1MPI](#)

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

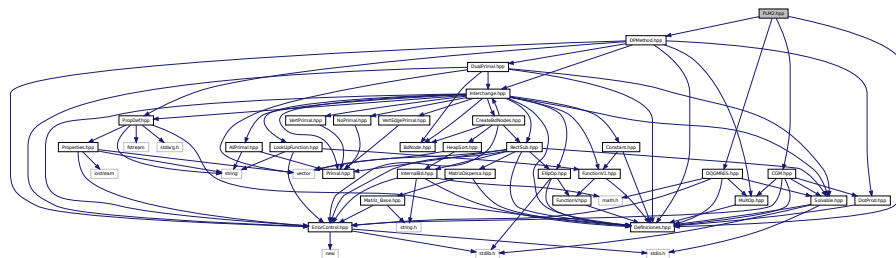
8.73 PLM2.cpp File Reference

```
#include "Definiciones.hpp"
#include "PLM2.hpp"
Include dependency graph for PLM2.cpp:
```

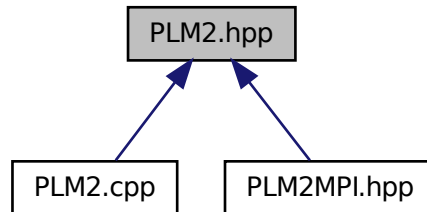


8.74 PLM2.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
Include dependency graph for PLM2.hpp:
```



This graph shows which files directly or indirectly include this file:



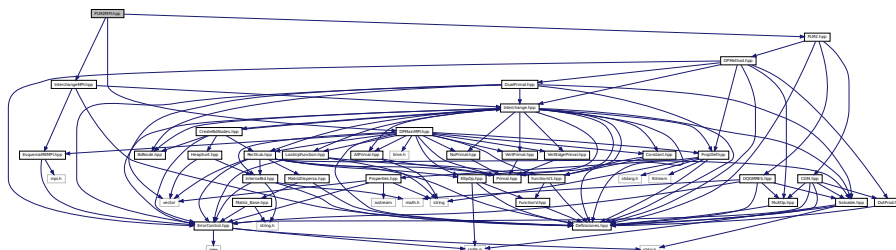
Classes

- class [PLM2](#)

8.75 PLM2MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PLM2.hpp"
```

Include dependency graph for PLM2MPI.hpp:



Classes

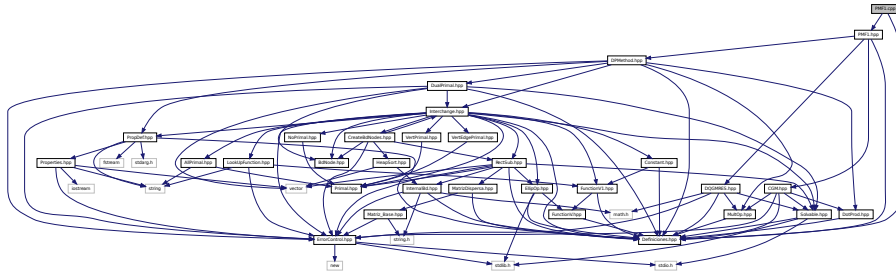
- class [PLM2MPI](#)

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

8.76 PMF1.cpp File Reference

```
#include "Definiciones.hpp"
#include "PMF1.hpp"
```

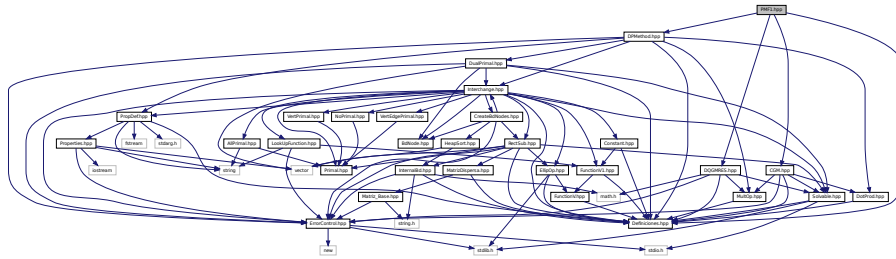
Include dependency graph for PMF1.cpp:



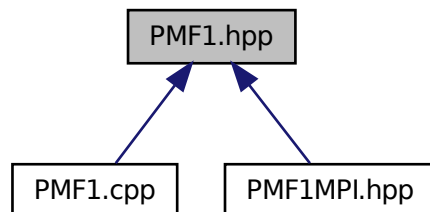
8.77 PMF1.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for PMF1.hpp:



This graph shows which files directly or indirectly include this file:

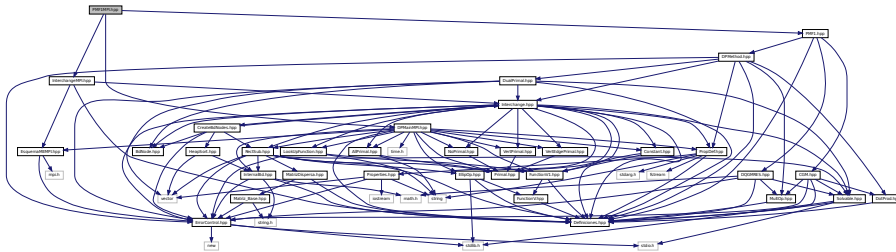


Classes

- class [PMF1](#)

8.78 PMF1MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PMF1.hpp"
Include dependency graph for PMF1MPI.hpp:
```



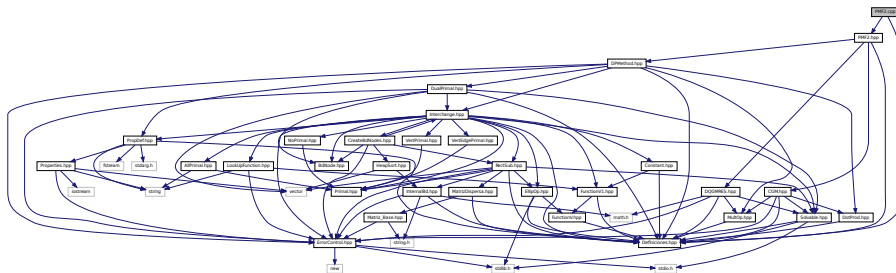
Classes

- 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:
```

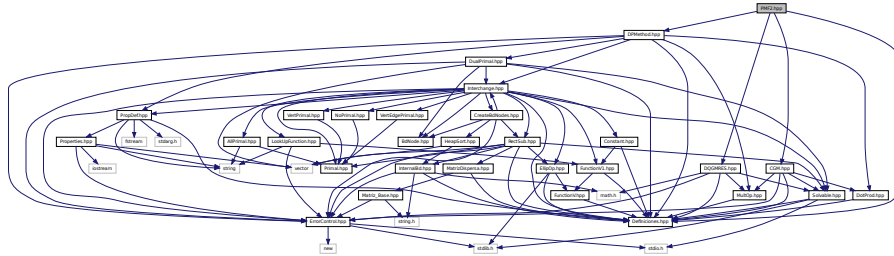


8.80 PMF2.hpp File Reference

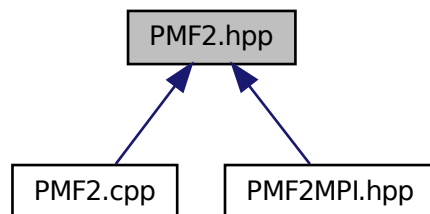
```
#include "Definiciones.hpp"
```



```
#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:



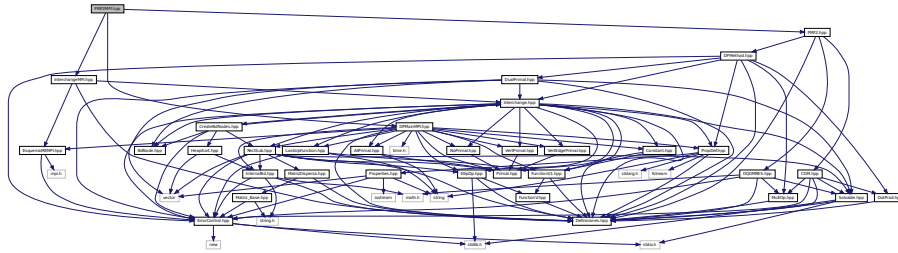
Classes

- class [PMF2](#)

8.81 PMF2MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PMF2.hpp"
```

Include dependency graph for PMF2MPI.hpp:



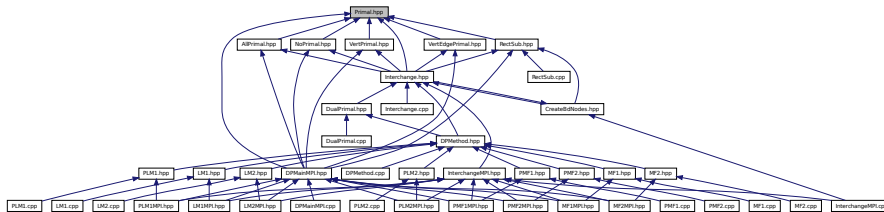
Classes

- 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:



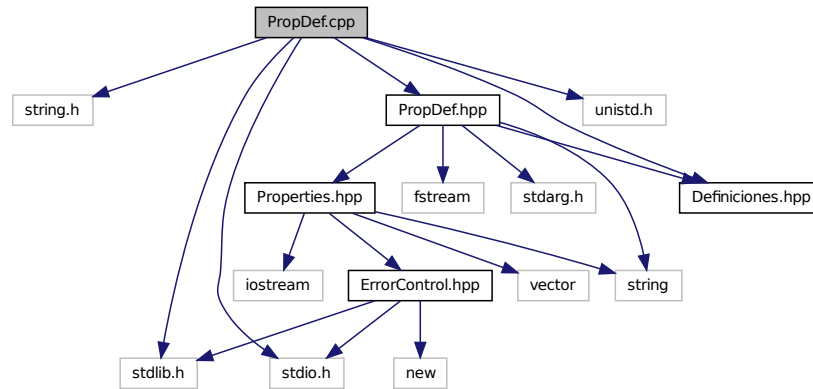
Classes

- 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:

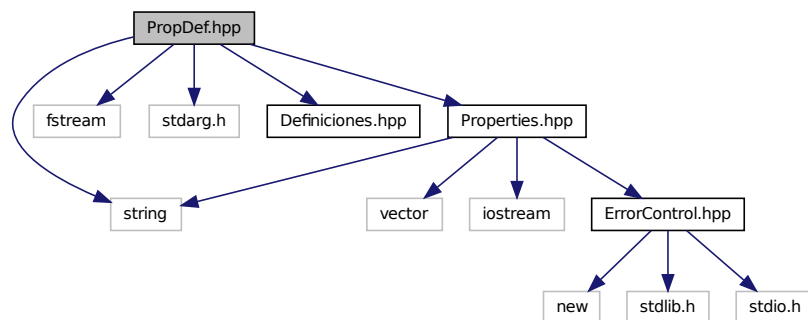


8.84 PropDef.hpp File Reference

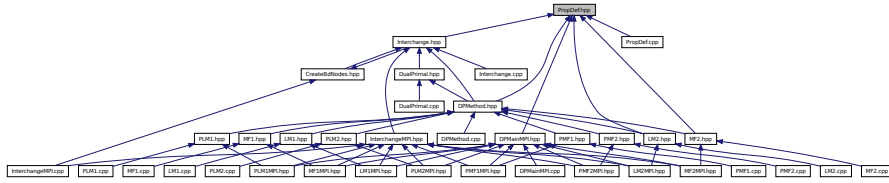
```

#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:

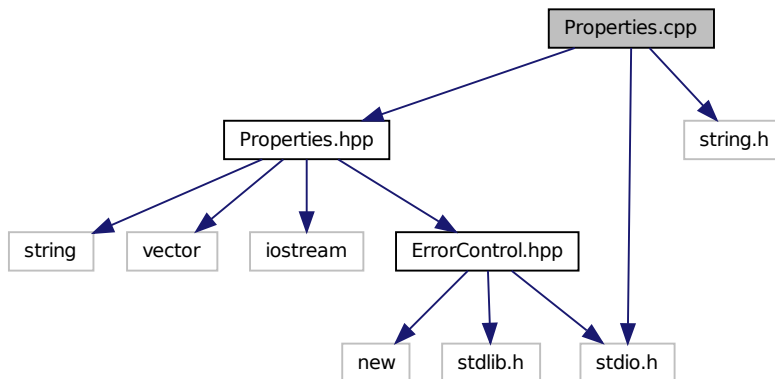


Classes

- class [PropDef](#)

8.85 Properties.cpp File Reference

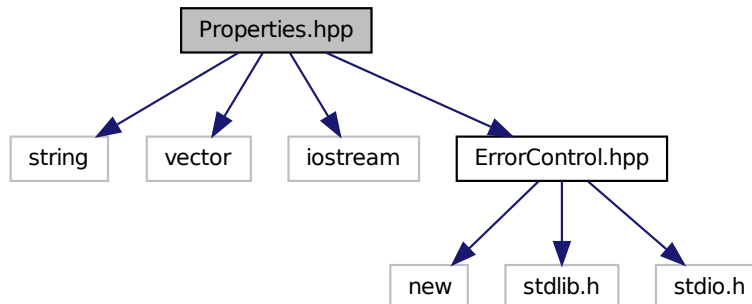
```
#include "Properties.hpp"
#include <string.h>
#include <stdio.h>
Include dependency graph for Properties.cpp:
```



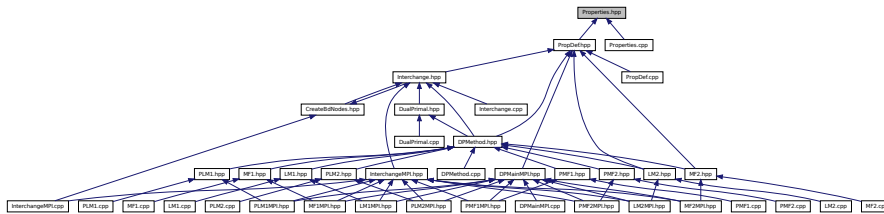
8.86 Properties.hpp File Reference

```
#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

```

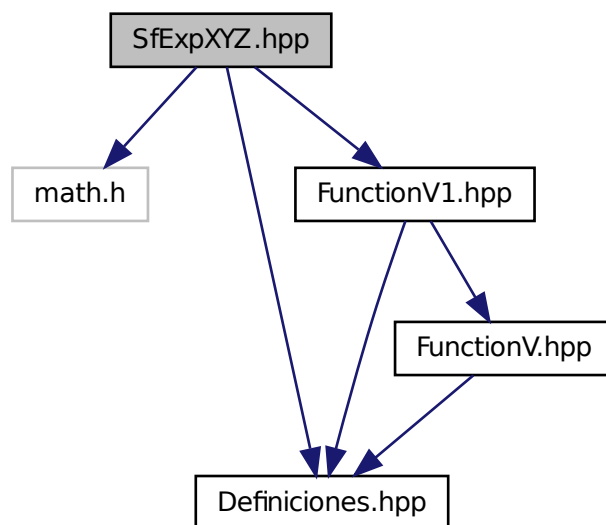
#include <stdlib.h>
#include "Definiciones.hpp"
#include "RectSub.hpp"
#include "BandSolve.hpp"
#include "BandCholesky.hpp"
#include "EllipOp.hpp"
#include "ICGM.hpp"
#include "IDQGMRES.hpp"
  
```


Classes

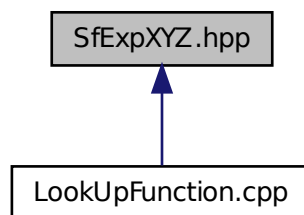
- class [RectSub](#)

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:

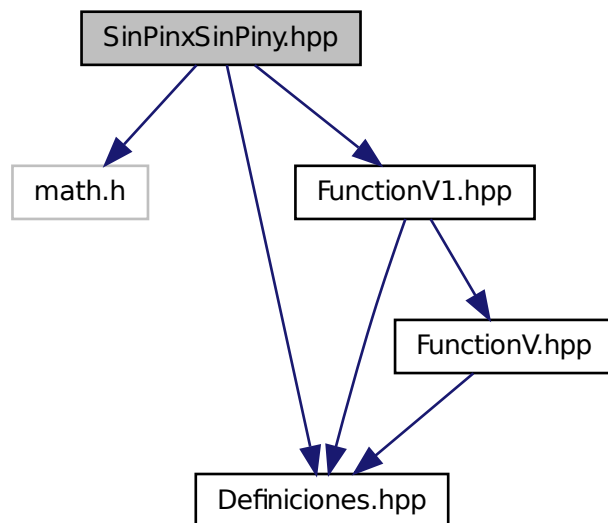


Classes

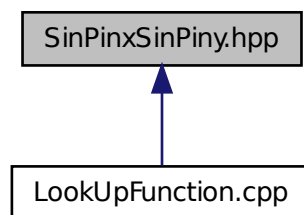
- class [SfExpXYZ](#)

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:

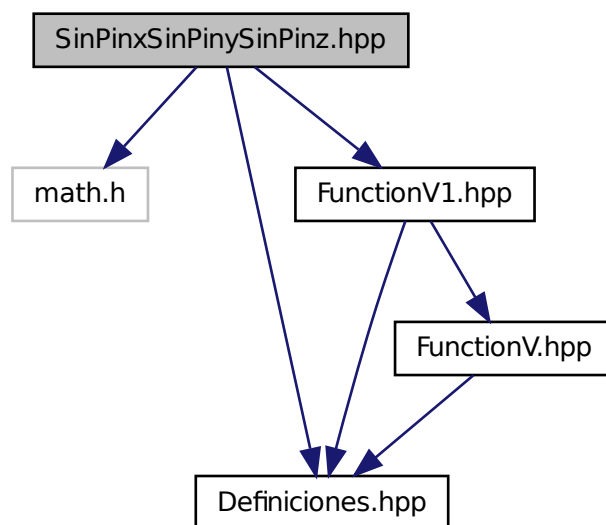


Classes

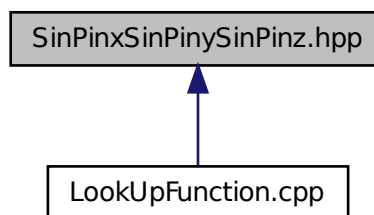
- class [SinPinxSinPiny](#)

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:

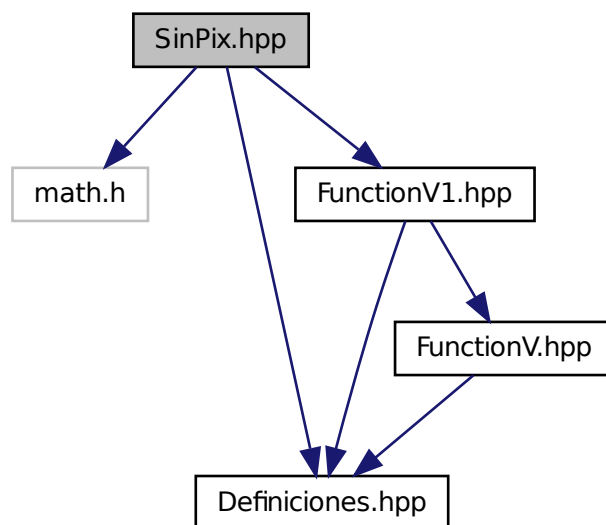


Classes

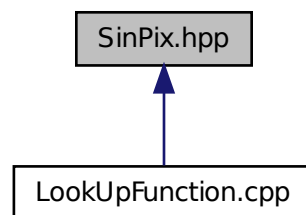
- class [SinPixSinPinySinPinz](#)

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:

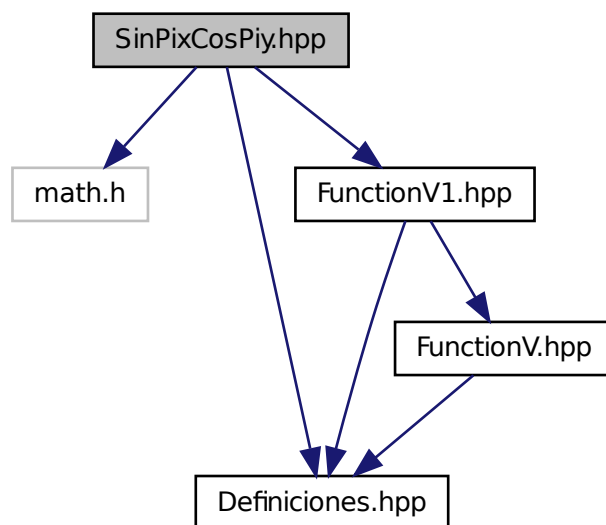


Classes

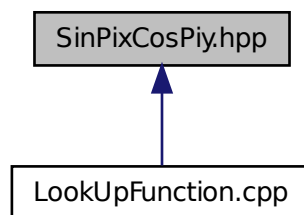
- class [SinPix](#)

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:

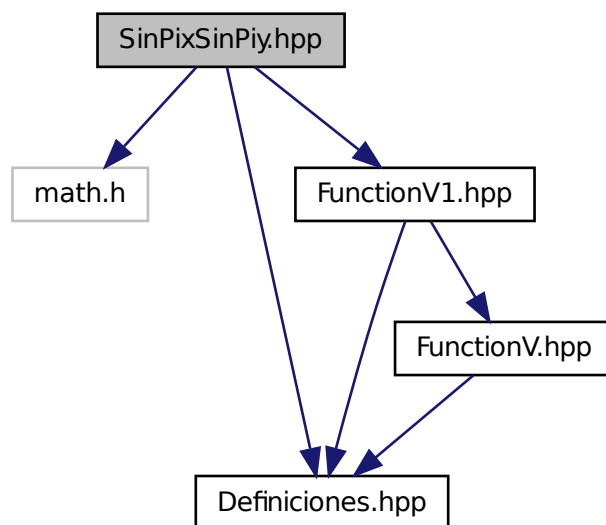


Classes

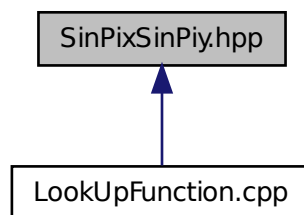
- class [SinPixCosPiy](#)

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:

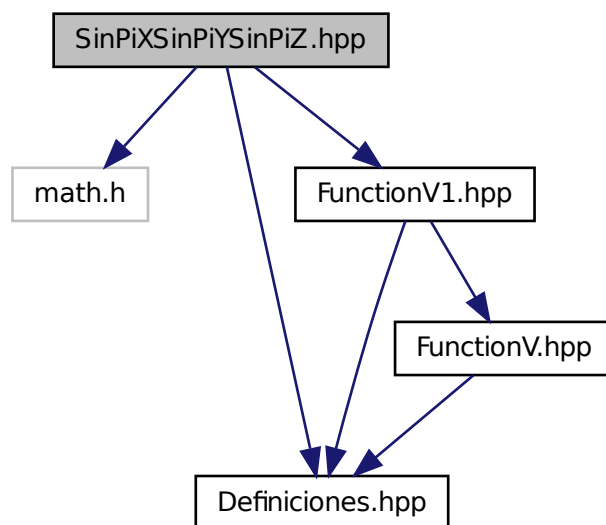


Classes

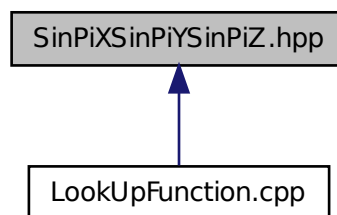
- class [SinPixSinPiy](#)

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:



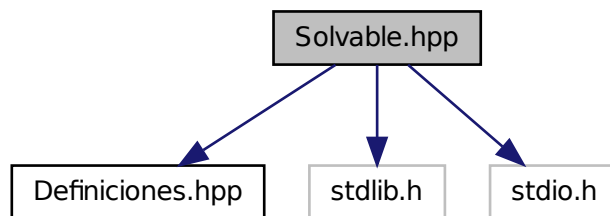
Classes

- class [SinPiXSinPiYSinPiZ](#)

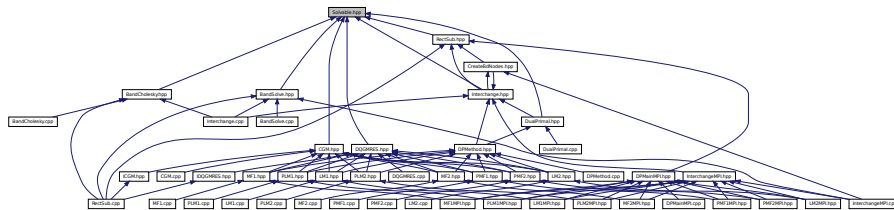
8.96 Solvable.hpp File Reference

```
#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:



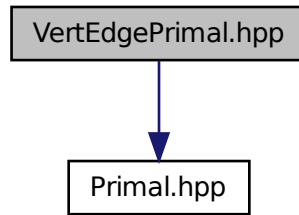
Classes

- class [Solvable](#)

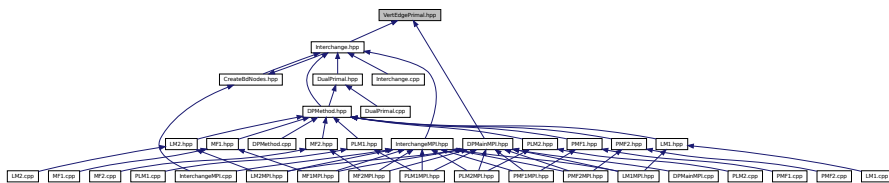
8.97 VertEdgePrimal.hpp File Reference

```
#include "Primal.hpp"
```

Include dependency graph for VertEdgePrimal.hpp:



This graph shows which files directly or indirectly include this file:



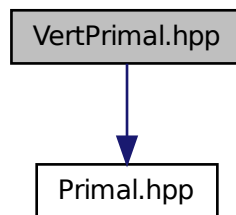
Classes

- class [VertEdgePrimal](#)

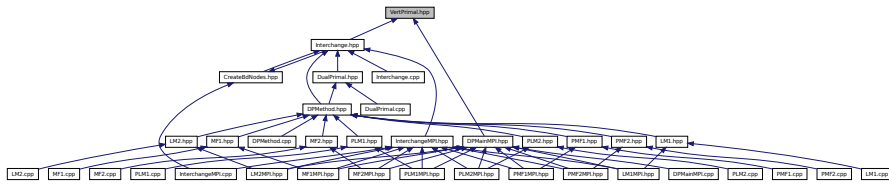
8.98 VertPrimal.hpp File Reference

```
#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

- ~BandCholesky
 - BandCholesky, [20](#)
- ~BandSolve
 - BandSolve, [24](#)
- ~CGM
 - CGM, [30](#)
- ~CreateBdNodes
 - CreateBdNodes, [36](#)
- ~DPMainMPI
 - DPMainMPI, [53](#)
- ~DPMethod
 - DPMethod, [57](#)
- ~DQGMRES
 - DQGMRES, [62](#)
- ~DualPrimal
 - DualPrimal, [65](#)
- ~EllipOp
 - EllipOp, [68](#)
- ~EsquemaMEMPI
 - EsquemaMEMPI, [77](#)
- ~FunctionV
 - FunctionV, [97](#)
- ~HeapSort
 - HeapSort, [101](#)
- ~ICGM
 - ICGM, [104](#)
- ~IDQGMRES
 - IDQGMRES, [108](#)
- ~Interchange
 - Interchange, [114](#)
- ~InternalBd
 - InternalBd, [126](#)
- ~MatrizDispersa
 - MatrizDispersa, [152](#)
- ~Matriz_Base
 - Matriz_Base, [146](#)
- ~Primal
 - Primal, [215](#)
- ~RectSub
 - RectSub, [225](#)
- ~Solvable
 - Solvable, [251](#)
- __Double__
 - Definiciones.hpp, [268](#)

A

- CGM, [31](#)
- a
 - Constant, [34](#)
 - DualPrimal, [65](#)
 - EllipOp, [68](#)
- AK
 - BandCholesky, [21](#)
 - BandSolve, [25](#)
 - MultBandSym, [173](#)
- addProjNs
 - RectSub, [225](#)
- AllPrimal, [15](#)
 - AllPrimal, [17](#)
 - AllPrimal, [17](#)
 - isPrimal, [17](#)
 - name, [17](#)
- AllPrimal.hpp, [259](#)
- alpha
 - CGM, [31](#)
- analyticSolution
 - DPMethod, [57](#)
- applyOmega
 - DQGMRES, [62](#)
- asigna
 - MatrizDispersa, [152](#)
- asignaNombre
 - Matriz_Base, [147](#)
- Ax
 - Interchange, [117](#)
- ax
 - Interchange, [117](#)
- Ay
 - Interchange, [117](#)
- ay
 - Interchange, [117](#)
- Az
 - Interchange, [117](#)
- az
 - Interchange, [117](#)
- b
 - EllipOp, [68](#)
- bFloat
 - RectSub, [226](#)
- Ban
 - Matriz_Base, [147](#)

- BandCholesky, 17
 - ~BandCholesky, 20
 - AK, 21
 - BandCholesky, 20
 - BandCholesky, 20
 - bw, 21
 - ce, 21
 - clean, 20
 - convertBand, 20
 - factorLU, 20
 - getIter, 20
 - n, 21
 - print, 20
 - solve, 20
- BandCholesky.cpp, 260
- BandCholesky.hpp, 260
- BandSolve, 21
 - ~BandSolve, 24
 - AK, 25
 - BandSolve, 24
 - BandSolve, 24
 - bw, 25
 - ce, 25
 - clean, 24
 - convertBand, 24
 - factorLU, 24
 - getIter, 24
 - n, 25
 - print, 24
 - solve, 25
- BandSolve.cpp, 262
- BandSolve.hpp, 262
- bd
 - InternalBd, 126
- bdAll
 - CreateBdNodes, 36
- bdDuals
 - CreateBdNodes, 36
- bdMap
 - RectSub, 226
- BdNode, 25
 - BdNode, 26
 - BdNode, 26
 - index, 26
 - mult, 26
 - node, 26
 - subd, 27
- BdNode.hpp, 264
- bdPrimals
 - CreateBdNodes, 36
- bdValues
 - Interchange, 117
- bds
 - Interchange, 117
- beta
 - CGM, 31
- bsym
 - RectSub, 226
- bw
 - BandCholesky, 21
 - BandSolve, 25
 - MultBandSym, 173
- bx
 - Interchange, 117
- by
 - Interchange, 117
- bz
 - Interchange, 117
- c
 - DPMainMPI, 53
 - EllipOp, 69
 - Interchange, 117
- CGM, 27
 - ~CGM, 30
 - A, 31
 - alpha, 31
 - beta, 31
 - CGM, 30
 - ce, 31
 - CGM, 30
 - clean, 30
 - dotP, 31
 - eps, 31
 - gamma, 31
 - getIter, 30
 - inicializa, 30
 - lambda, 31
 - mu, 31
 - n, 31
 - nIter, 31
 - nMaxIter, 31
 - norm, 31
 - p, 31
 - r, 31
 - setEpsilon, 31
 - setMaxIter, 31
 - solve, 31
 - v, 31
- CGM.cpp, 264
- CGM.hpp, 265
- Ca
 - RectSub, 226
- calcSP
 - Interchange, 114
- calcValues
 - DualPrimal, 65
- calcula

- Interchange, [114](#)
- InterchangeMPI, [122](#)
- Cb
 - RectSub, [226](#)
- Cc
 - RectSub, [227](#)
- ce
 - BandCholesky, [21](#)
 - BandSolve, [25](#)
 - CGM, [31](#)
 - CreateBdNodes, [36](#)
 - DPMMethod, [58](#)
 - DQGMRES, [63](#)
 - DualPrimal, [66](#)
 - EsquemaMEMPI, [77](#)
 - Interchange, [117](#)
 - LookUpFunction, [143](#)
 - Matriz_Base, [147](#)
 - Properties, [221](#)
 - RectSub, [227](#)
- clean
 - BandCholesky, [20](#)
 - BandSolve, [24](#)
 - CGM, [30](#)
 - DPMMethod, [58](#)
 - DQGMRES, [62](#)
 - ICGM, [105](#)
 - IDQGMRES, [108](#)
 - LM1, [130](#)
 - LM1MPI, [134](#)
 - LM2, [137](#)
 - LM2MPI, [141](#)
 - MF1, [158](#)
 - MF1MPI, [162](#)
 - MF2, [166](#)
 - MF2MPI, [170](#)
 - PLM1, [186](#)
 - PLM1MPI, [190](#)
 - PLM2, [193](#)
 - PLM2MPI, [197](#)
 - PMF1, [200](#)
 - PMF1MPI, [204](#)
 - PMF2, [208](#)
 - PMF2MPI, [212](#)
 - Solvable, [251](#)
- clear
 - Interchange, [114](#)
 - InterchangeMPI, [122](#)
 - RectSub, [225](#)
- coef
 - RectSub, [227](#)
- Col
 - Matriz_Base, [147](#)
- columnas
 - Matriz_Base, [147](#)
- compareTo
 - InternalBd, [126](#)
- conditionalNumber
 - DPMMethod, [58](#)
- Constant, [32](#)
 - a, [34](#)
 - Constant, [34](#)
 - eval, [34](#)
 - getVar, [34](#)
 - setVar, [34](#)
- Constant.hpp, [265](#)
- convertBand
 - BandCholesky, [20](#)
 - BandSolve, [24](#)
- coord
 - InternalBd, [126](#)
 - RectSub, [227](#)
- coordN
 - RectSub, [227](#)
- CreateBdNodes, [34](#)
 - ~CreateBdNodes, [36](#)
 - bdAll, [36](#)
 - bdDuals, [36](#)
 - bdPrimals, [36](#)
 - ce, [36](#)
 - CreateBdNodes, [36](#)
 - CreateBdNodes, [36](#)
 - dualMult, [36](#)
 - hbd, [36](#)
 - ibd, [36](#)
 - ibdAll, [36](#)
 - maxBd, [36](#)
 - nD, [36](#)
 - nDual, [36](#)
 - nP, [36](#)
 - nPrimal, [36](#)
- CreateBdNodes.hpp, [266](#)
- cs
 - DQGMRES, [63](#)
- DIM_VECTOR
 - Definiciones.hpp, [268](#)
- DPMainMPI, [50](#)
 - ~DPMainMPI, [53](#)
 - c, [53](#)
 - DPMainMPI, [53](#)
 - deleteInternalBd, [53](#)
 - domain, [53](#)
 - DPMainMPI, [53](#)
 - Esclavo, [53](#)
 - f, [53](#)
 - fc, [53](#)
 - g, [53](#)

- gc, [53](#)
- hbd, [53](#)
- indl, [53](#)
- mesh, [53](#)
- method, [54](#)
- msa, [54](#)
- mss, [54](#)
- nDim, [54](#)
- nOmega, [54](#)
- nta, [54](#)
- omegas, [54](#)
- one, [54](#)
- op, [54](#)
- prim, [54](#)
- primal, [54](#)
- sf, [54](#)
- sg, [54](#)
- swprint, [54](#)
- t1, [54](#)
- t2, [54](#)
- xnp, [54](#)
- zero, [54](#)
- DPMainMPI.cpp, [274](#)
- DPMainMPI.hpp, [274](#)
- DPMMethod, [55](#)
 - ~DPMMethod, [57](#)
 - analyticSolution, [57](#)
 - ce, [58](#)
 - clean, [58](#)
 - conditionalNumber, [58](#)
 - DPMMethod, [57](#)
 - DPMMethod, [57](#)
 - dualp, [58](#)
 - epsilon, [59](#)
 - genInverse, [58](#)
 - getSize, [58](#)
 - iniInterchage, [58](#)
 - initialize, [58](#)
 - inter, [59](#)
 - nDim, [59](#)
 - nDual, [59](#)
 - nOmega, [59](#)
 - prCoord, [58](#)
 - print, [58](#)
 - printTime, [58](#)
 - printv, [59](#)
 - props, [59](#)
 - rhs, [58](#)
 - rhss, [59](#)
 - scr, [59](#)
 - solve, [58](#)
 - solver, [59](#)
 - time0, [59](#)
 - time1, [59](#)
 - time2, [59](#)
 - time3, [59](#)
 - u, [59](#)
- DPMMethod.cpp, [275](#)
- DPMMethod.hpp, [275](#)
- DQGMRES, [59](#)
 - ~DQGMRES, [62](#)
 - applyOmega, [62](#)
 - ce, [63](#)
 - clean, [62](#)
 - cs, [63](#)
 - DQGMRES, [62](#)
 - DQGMRES, [62](#)
 - eps, [63](#)
 - getIter, [62](#)
 - gm, [63](#)
 - gm1, [63](#)
 - h, [63](#)
 - inicializa, [62](#)
 - k, [63](#)
 - k1, [63](#)
 - maxIter, [63](#)
 - mult, [63](#)
 - n, [63](#)
 - nIter, [63](#)
 - nMaxIter, [63](#)
 - p, [63](#)
 - q, [63](#)
 - setEpsilon, [63](#)
 - setMaxIter, [63](#)
 - solve, [63](#)
 - v, [63](#)
- DQGMRES.cpp, [276](#)
- DQGMRES.hpp, [277](#)
- DUAL
 - RectSub, [227](#)
- Definiciones.hpp, [267](#)
 - __Double__, [268](#)
 - DIM_VECTOR, [268](#)
 - EPS_EQUAL, [268](#)
 - EPSILON, [268](#)
 - EPSILON_LOCAL, [268](#)
 - ldouble, [269](#)
 - NMAXITER, [268](#)
 - NMAXITER_LOCAL, [268](#)
 - RESIDUAL, [269](#)
- deleteInternalBd
 - DPMainMPI, [53](#)
- diff
 - Interchange, [114](#)
 - InterchangeMPI, [122](#)
 - RectSub, [225](#)
- diffValues
 - Interchange, [114](#)

- InterchangeMPI, 122
- RectSub, 225
- dim
 - FunctionV, 98
- dimension
 - FunctionV, 97
- Disc, 37
 - Disc, 39
 - eval, 39
 - getVar, 39
 - setVar, 39
 - var, 39
- Disc.hpp, 269
- Disc13, 39
 - Disc13, 42
 - eval, 42
 - getVar, 42
 - setVar, 42
 - var, 42
- Disc13.hpp, 270
- Disc14, 42
 - Disc14, 45
 - eval, 45
 - getVar, 45
 - setVar, 45
 - var, 45
- Disc14.hpp, 271
- Disc15, 45
 - Disc15, 48
 - eval, 48
 - getVar, 48
 - setVar, 48
 - var, 48
- Disc15.hpp, 272
- domain
 - DPMainMPI, 53
 - Interchange, 117
 - RectSub, 227
- dot
 - DotProd, 49
 - ICGM, 105
 - LM1, 130
 - LM2, 137
 - MF1, 158
 - MF2, 166
 - PLM1, 186
 - PLM2, 193
 - PMF1, 200
 - PMF2, 208
- dotP
 - CGM, 31
- DotProd, 48
 - dot, 49
- DotProd.hpp, 273
- dp
 - InternalBd, 126
- dualMult
 - CreateBdNodes, 36
- DualPrimal, 64
 - ~DualPrimal, 65
 - a, 65
 - calcValues, 65
 - ce, 66
 - DualPrimal, 65
 - DualPrimal, 65
 - fromSubdomains, 65
 - genMats, 65
 - getNDual, 65
 - inter, 66
 - j, 65
 - multS, 65
 - nD, 66
 - nDual, 66
 - nP, 66
 - nPrimal, 66
 - SP, 66
 - solveAPP, 65
 - solveS, 66
 - toSubdomains, 66
 - XP, 66
 - YP, 66
- DualPrimal.cpp, 277
- DualPrimal.hpp, 278
- dualp
 - DPMethod, 58
- EDGE
 - Primal, 215
 - RectSub, 227
- EPS_EQUAL
 - Definiciones.hpp, 268
- EPSILON
 - Definiciones.hpp, 268
- EPSILON_LOCAL
 - Definiciones.hpp, 268
- EllipOp, 66
 - ~EllipOp, 68
 - a, 68
 - b, 68
 - c, 69
 - EllipOp, 68
 - EllipOp, 68
 - f, 69
 - g, 69
 - getA, 68
 - getB, 68
 - getC, 68
 - getF, 68

- getG, [68](#)
- isSymmetric, [68](#)
- nDim, [69](#)
- setF, [68](#)
- setG, [68](#)
- sol, [69](#)
- EllipOp.hpp, [279](#)
- eps
 - CGM, [31](#)
 - DQGMRES, [63](#)
- epsilon
 - DPMMethod, [59](#)
- equals
 - InternalBd, [126](#)
- ErrorControl, [69](#)
 - ErrorControl, [71](#)
 - ErrorControl, [71](#)
 - fatalError, [72](#)
 - memoryError, [72](#)
 - nameClass, [72](#)
 - nameClassFunct, [73](#)
 - nameFunct, [73](#)
 - nmClass, [73](#)
 - nmFunction, [73](#)
- ErrorControl.cpp, [279](#)
- ErrorControl.hpp, [280](#)
- Esclavo
 - DPMMainMPI, [53](#)
- EsquemaMEMPI, [73](#)
 - ~EsquemaMEMPI, [77](#)
 - ce, [77](#)
 - EsquemaMEMPI, [77](#)
 - EsquemaMEMPI, [77](#)
 - generaRepartoCarga, [77](#)
 - id, [77](#)
 - np, [78](#)
 - npu, [78](#)
 - numeroProcesadoresUsar, [77](#)
 - reparteCargaTrabajo, [77](#)
 - ta, [78](#)
- EsquemaMEMPI.cpp, [281](#)
- EsquemaMEMPI.hpp, [282](#)
- eval
 - Constant, [34](#)
 - Disc, [39](#)
 - Disc13, [42](#)
 - Disc14, [45](#)
 - Disc15, [48](#)
 - ExpVXY, [81](#)
 - ExpVXYZ, [84](#)
 - ExpX, [87](#)
 - ExpXY, [90](#)
 - ExpXYZ, [93](#)
 - fExpXY, [96](#)
 - FunctionV, [97](#)
 - NSfExpXY, [180](#)
 - NSfExpXYZ, [183](#)
 - SfExpXYZ, [231](#)
 - SinPinxSinPiny, [234](#)
 - SinPinxSinPinySinPinz, [237](#)
 - SinPix, [240](#)
 - SinPixCosPiy, [243](#)
 - SinPixSinPiy, [246](#)
 - SinPiXSinPiYSinPiZ, [249](#)
- ExpVXY, [78](#)
 - eval, [81](#)
 - ExpVXY, [81](#)
 - ExpVXY, [81](#)
 - getVar, [81](#)
 - setVar, [81](#)
 - var, [81](#)
- ExpVXY.hpp, [282](#)
- ExpVXYZ, [81](#)
 - eval, [84](#)
 - ExpVXYZ, [84](#)
 - ExpVXYZ, [84](#)
 - getVar, [84](#)
 - setVar, [84](#)
 - var, [84](#)
- ExpVXYZ.hpp, [283](#)
- ExpX, [84](#)
 - eval, [87](#)
 - ExpX, [87](#)
 - ExpX, [87](#)
 - getVar, [87](#)
 - setVar, [87](#)
 - var, [87](#)
- ExpX.hpp, [285](#)
- ExpXY, [87](#)
 - eval, [90](#)
 - ExpXY, [90](#)
 - ExpXY, [90](#)
 - getVar, [90](#)
 - setVar, [90](#)
 - var, [90](#)
- ExpXY.hpp, [286](#)
- ExpXYZ, [90](#)
 - eval, [93](#)
 - ExpXYZ, [93](#)
 - ExpXYZ, [93](#)
 - getVar, [93](#)
 - setVar, [93](#)
 - var, [93](#)
- ExpXYZ.hpp, [287](#)
- f
 - DPMMainMPI, [53](#)
 - EllipOp, [69](#)

- Interchange, 117
 - LM1, 131
 - LM2, 138
 - PLM1, 187
 - PLM2, 194
- FACE
 - Primal, 215
 - RectSub, 227
- fExpXY, 93
 - eval, 96
 - fExpXY, 96
 - fExpXY, 96
 - getVar, 96
 - setVar, 96
 - var, 96
- fExpXY.hpp, 288
- factorLU
 - BandCholesky, 20
 - BandSolve, 24
- fatalError
 - ErrorControl, 72
- fc
 - DPMainMPI, 53
 - Interchange, 117
- fromSubdomains
 - DualPrimal, 65
 - Interchange, 114
 - InterchangeMPI, 122
- FunctionV, 96
 - ~FunctionV, 97
 - dim, 98
 - dimension, 97
 - eval, 97
 - FunctionV, 97
 - FunctionV, 97
- FunctionV.hpp, 289
- FunctionV1, 98
 - getVar, 99
 - setVar, 99
- FunctionV1.hpp, 289
- g
 - DPMainMPI, 53
 - EllipOp, 69
 - Interchange, 117
- gamma
 - CGM, 31
- gc
 - DPMainMPI, 53
 - Interchange, 117
- genCoef
 - RectSub, 225
- genCoefVar
 - RectSub, 225
- genGeom
 - Interchange, 114
- genHeap
 - HeapSort, 101
- genInv
 - Interchange, 115
 - InterchangeMPI, 122
 - RectSub, 225
- genInverse
 - DPMMethod, 58
 - RectSub, 225
- genMats
 - DualPrimal, 65
- genNcoord
 - RectSub, 225
- genNtype
 - RectSub, 225
- generaRepartoCarga
 - EsquemaMEMPI, 77
- getA
 - EllipOp, 68
- getB
 - EllipOp, 68
- getBdSize
 - RectSub, 225
- getC
 - EllipOp, 68
- getCoord
 - RectSub, 225
- getCoordNode
 - Interchange, 115
 - InterchangeMPI, 122
 - RectSub, 225
- getDouble
 - PropDef, 218
- getF
 - EllipOp, 68
 - LookUpFunction, 143
- getG
 - EllipOp, 68
- getInt
 - PropDef, 218
- getInternalBd
 - InterchangeMPI, 122
 - RectSub, 225
- getInv
 - RectSub, 225
- getIter
 - BandCholesky, 20
 - BandSolve, 24
 - CGM, 30
 - DQGMRES, 62
 - Solvable, 252
- getMaxBdSize

- Interchange, 115
- InterchangeMPI, 122
- getND
 - Interchange, 115
- getNDual
 - DualPrimal, 65
- getNDuals
 - Interchange, 115
- getNP
 - Interchange, 115
 - RectSub, 225
- getNPPrimals
 - Interchange, 115
- getName
 - Solvable, 252
- getNtype
 - Interchange, 115
 - InterchangeMPI, 123
 - RectSub, 225
- getPrimals
 - Interchange, 115
 - InterchangeMPI, 123
 - RectSub, 225
- getProperty
 - Properties, 221
- getSize
 - DPMMethod, 58
 - ICGM, 105
 - IDQGMRES, 108
 - MultBandSym, 172
 - MultOp, 175
- getString
 - PropDef, 218
- getValue
 - Interchange, 115
 - InterchangeMPI, 123
 - RectSub, 225
- getValues
 - RectSub, 225
- getVar
 - Constant, 34
 - Disc, 39
 - Disc13, 42
 - Disc14, 45
 - Disc15, 48
 - ExpVXY, 81
 - ExpVXYZ, 84
 - ExpX, 87
 - ExpXY, 90
 - ExpXYZ, 93
 - fExpXY, 96
 - FunctionV1, 99
 - NSfExpXY, 180
 - NSfExpXYZ, 183
 - SfExpXYZ, 231
 - SinPinxSinPiny, 234
 - SinPinxSinPinySinPinz, 237
 - SinPix, 240
 - SinPixCosPiy, 243
 - SinPixSinPiy, 246
 - SinPiXSinPiYSinPiZ, 249
- getnDim
 - Interchange, 115
- getnOmega
 - Interchange, 115
- getval
 - InternalBd, 126
- gm
 - DQGMRES, 63
- gm1
 - DQGMRES, 63
- h
 - DQGMRES, 63
 - RectSub, 227
- hbd
 - CreateBdNodes, 36
 - DPMMainMPI, 53
- HeapSort, 100
 - ~HeapSort, 101
- genHeap, 101
- HeapSort, 101
- HeapSort, 101
- n, 101
- r, 101
- rr, 101
- siftup, 101
- sort, 101
- swap, 101
- HeapSort.hpp, 290
- hfac
 - RectSub, 227
- i
 - ICGM, 105
- ICGM, 101
 - ~ICGM, 104
- clean, 105
- dot, 105
- getSize, 105
- i, 105
- ICGM, 104
- ICGM, 104
- M, 105
- multOp, 105
- val, 105
- ICGM.hpp, 291
- IDQGMRES, 105
 - ~IDQGMRES, 108

- clean, 108
- getSize, 108
- IDQGMRES, 108
- IDQGMRES, 108
- M, 109
- multOp, 109
- IDQGMRES.hpp, 292
- INTBD
 - Primal, 215
 - RectSub, 227
- INTERIOR
 - Primal, 215
 - RectSub, 227
- ibd
 - CreateBdNodes, 36
- ibdAll
 - CreateBdNodes, 36
- id
 - EsquemaMEMPI, 77
 - RectSub, 227
- index
 - BdNode, 26
- incl
 - DPMainMPI, 53
 - InterchangeMPI, 124
- iniInterchage
 - DPMethod, 58
 - LM1MPI, 134
 - LM2MPI, 141
 - MF1MPI, 162
 - MF2MPI, 170
 - PLM1MPI, 190
 - PLM2MPI, 197
 - PMF1MPI, 204
 - PMF2MPI, 212
- inicializa
 - CGM, 30
 - DQGMRES, 62
 - MatrizDispersa, 152
- initialize
 - DPMethod, 58
 - Interchange, 115
- inter
 - DPMethod, 59
 - DualPrimal, 66
- Interchange, 109
 - ~Interchange, 114
 - Ax, 117
 - ax, 117
 - Ay, 117
 - ay, 117
 - Az, 117
 - az, 117
 - bdValues, 117
 - bds, 117
 - bx, 117
 - by, 117
 - bz, 117
 - c, 117
 - calcSP, 114
 - calcula, 114
 - ce, 117
 - clear, 114
 - diff, 114
 - diffValues, 114
 - domain, 117
 - f, 117
 - fc, 117
 - fromSubdomains, 114
 - g, 117
 - gc, 117
 - genGeom, 114
 - genInv, 115
 - getCoordNode, 115
 - getMaxBdSize, 115
 - getND, 115
 - getNDuals, 115
 - getNP, 115
 - getNPrimals, 115
 - getNtype, 115
 - getPrimals, 115
 - getValue, 115
 - getnDim, 115
 - getnOmega, 115
 - initialize, 115
 - Interchange, 114
 - inverse, 115
 - knownValues, 116
 - mesh, 117
 - method, 117
 - multOp, 116
 - nDim, 117
 - nOmega, 118
 - omegas, 118
 - one, 118
 - op, 118
 - pbdValues, 116
 - prim, 118
 - primal, 118
 - print, 116
 - props, 118
 - rbdValues, 116
 - rhs, 116
 - sbdValues, 116
 - setNtype, 116
 - setPrimals, 116
 - setValue, 116
 - sf, 118

- sg, 118
- swprint, 118
- toSubdomains, 117
- zero, 118
- Interchange.cpp, 293
- Interchange.hpp, 294
- InterchangeMPI, 118
 - calcula, 122
 - clear, 122
 - diff, 122
 - diffValues, 122
 - fromSubdomains, 122
 - genInv, 122
 - getCoordNode, 122
 - getInternalBd, 122
 - getMaxBdSize, 122
 - getNtype, 123
 - getPrimals, 123
 - getValue, 123
 - indl, 124
 - InterchangeMPI, 122
 - InterchangeMPI, 122
 - inverse, 123
 - knownValues, 123
 - ME, 124
 - msa, 124
 - mss, 124
 - multOp, 123
 - print, 123
 - rhs, 123
 - setNtype, 123
 - setPrimals, 124
 - toSubdomains, 124
 - xnp, 124
- InterchangeMPI.cpp, 295
- InterchangeMPI.hpp, 295
- InternalBd, 124
 - ~InternalBd, 126
 - bd, 126
 - compareTo, 126
 - coord, 126
 - dp, 126
 - equals, 126
 - getval, 126
 - InternalBd, 126
 - InternalBd, 126
 - nDim, 126
 - node, 126
 - rbd, 126
 - rcoord, 126
 - rdp, 126
 - rnode, 126
 - rsubd, 126
 - subd, 126
- InternalBd.hpp, 296
- inv
 - RectSub, 227
- inverse
 - Interchange, 115
 - InterchangeMPI, 123
 - RectSub, 225
- isDual
 - RectSub, 225
- isFloat
 - RectSub, 226
- isIntBd
 - RectSub, 226
- isInterior
 - RectSub, 226
- isKnown
 - RectSub, 226
- isPrimal
 - AllPrimal, 17
 - NoPrimal, 177
 - Primal, 215
 - RectSub, 226
 - VertEdgePrimal, 255
 - VertPrimal, 258
- isSymmetric
 - EllipOp, 68
- isVertex
 - RectSub, 226
- J
 - MatrizDispersa, 155
- j
 - DualPrimal, 65
- k
 - DQGMRES, 63
- k1
 - DQGMRES, 63
- KNOWN
 - Primal, 215
 - RectSub, 227
- keys
 - Properties, 221
- knownValues
 - Interchange, 116
 - InterchangeMPI, 123
 - RectSub, 226
- LM1, 127
 - clean, 130
 - dot, 130
 - f, 131
 - LM1, 130
 - LM1, 130
 - multOp, 130

- op, [131](#)
 - rhs, [130](#)
 - solve, [130](#)
 - uf, [131](#)
 - up, [131](#)
- LM1.cpp, [296](#)
- LM1.hpp, [297](#)
- LM1MPI, [131](#)
 - clean, [134](#)
 - iniInterchage, [134](#)
 - LM1MPI, [134](#)
 - LM1MPI, [134](#)
 - solvePar, [134](#)
- LM1MPI.hpp, [298](#)
- LM2, [135](#)
 - clean, [137](#)
 - dot, [137](#)
 - f, [138](#)
 - LM2, [137](#)
 - LM2, [137](#)
 - multOp, [137](#)
 - op, [138](#)
 - rhs, [137](#)
 - solve, [137](#)
 - uf, [138](#)
 - ug, [138](#)
 - up, [138](#)
- LM2.cpp, [298](#)
- LM2.hpp, [298](#)
- LM2MPI, [138](#)
 - clean, [141](#)
 - iniInterchage, [141](#)
 - LM2MPI, [141](#)
 - LM2MPI, [141](#)
 - solvePar, [141](#)
- LM2MPI.hpp, [299](#)
- lambda
 - CGM, [31](#)
- ldouble
 - Definiciones.hpp, [269](#)
- list
 - Properties, [221](#)
- load
 - Properties, [221](#)
- LookUpFunction, [142](#)
 - ce, [143](#)
 - getF, [143](#)
- LookUpFunction.cpp, [300](#)
- LookUpFunction.hpp, [301](#)
- M
 - ICGM, [105](#)
 - IDQGMRES, [109](#)
 - MatrizDispersa, [155](#)
 - RectSub, [227](#)
- M1
 - RectSub, [227](#)
- ME
 - InterchangeMPI, [124](#)
- MF1, [155](#)
 - clean, [158](#)
 - dot, [158](#)
 - MF1, [158](#)
 - MF1, [158](#)
 - multOp, [158](#)
 - op, [158](#)
 - rhs, [158](#)
 - solve, [158](#)
- MF1.cpp, [304](#)
- MF1.hpp, [305](#)
- MF1MPI, [159](#)
 - clean, [162](#)
 - iniInterchage, [162](#)
 - MF1MPI, [162](#)
 - MF1MPI, [162](#)
 - solvePar, [162](#)
- MF1MPI.hpp, [306](#)
- MF2, [163](#)
 - clean, [166](#)
 - dot, [166](#)
 - MF2, [166](#)
 - MF2, [166](#)
 - multOp, [166](#)
 - op, [166](#)
 - rhs, [166](#)
 - solve, [166](#)
 - up, [166](#)
- MF2.cpp, [306](#)
- MF2.hpp, [306](#)
- MF2MPI, [167](#)
 - clean, [170](#)
 - iniInterchage, [170](#)
 - MF2MPI, [170](#)
 - MF2MPI, [170](#)
 - solvePar, [170](#)
- MF2MPI.hpp, [307](#)
- main.hpp, [301](#)
- mapFull
 - RectSub, [227](#)
- mapInt
 - RectSub, [227](#)
- Matriz_Base, [143](#)
 - ~Matriz_Base, [146](#)
 - asignaNombre, [147](#)
 - Ban, [147](#)
 - ce, [147](#)
 - Col, [147](#)
 - columnas, [147](#)

- Matriz_Base, [146](#)
- Matriz_Base, [146](#)
- Nmb, [147](#)
- Ren, [147](#)
- renglones, [147](#)
- Matriz_Base.hpp, [301](#)
- MatrizDispersa, [148](#)
 - ~MatrizDispersa, [152](#)
 - asigna, [152](#)
 - inicializa, [152](#)
 - J, [155](#)
 - M, [155](#)
 - MatrizDispersa, [152](#)
 - MatrizDispersa, [152](#)
 - multiplica, [153](#)
 - retorna, [153](#)
 - retornaNumeroColumna, [153](#)
 - retornaNumeroColumnasBanda, [153](#)
 - retornaValorColumna, [154](#)
 - solicitaMemoria, [154](#)
 - tamanoBanda, [154](#)
 - visualiza, [154](#)
 - visualizaMatricesInternas, [154](#)
- MatrizDispersa.cpp, [302](#)
- MatrizDispersa.hpp, [303](#)
- maxBd
 - CreateBdNodes, [36](#)
- maxIter
 - DQGMRES, [63](#)
- memoryError
 - ErrorControl, [72](#)
- mesh
 - DPMainMPI, [53](#)
 - Interchange, [117](#)
 - RectSub, [227](#)
- method
 - DPMainMPI, [54](#)
 - Interchange, [117](#)
- msa
 - DPMainMPI, [54](#)
 - InterchangeMPI, [124](#)
- mss
 - DPMainMPI, [54](#)
 - InterchangeMPI, [124](#)
- mu
 - CGM, [31](#)
- mult
 - BdNode, [26](#)
 - DQGMRES, [63](#)
- MultBandSym, [171](#)
 - AK, [173](#)
 - bw, [173](#)
 - getSize, [172](#)
 - MultBandSym, [172](#)
 - multOp, [173](#)
 - MultBandSym, [172](#)
 - n, [173](#)
- MultBandSym.cpp, [308](#)
- MultBandSym.hpp, [309](#)
- MultOp, [173](#)
 - getSize, [175](#)
 - multOp, [175](#)
- multOp
 - ICGM, [105](#)
 - IDQGMRES, [109](#)
 - Interchange, [116](#)
 - InterchangeMPI, [123](#)
 - LM1, [130](#)
 - LM2, [137](#)
 - MF1, [158](#)
 - MF2, [166](#)
 - MultBandSym, [173](#)
 - MultOp, [175](#)
 - PLM1, [186](#)
 - PLM2, [193](#)
 - PMF1, [200](#)
 - PMF2, [208](#)
 - RectSub, [226](#)
- MultOp.hpp, [309](#)
- multS
 - DualPrimal, [65](#)
- multiplica
 - MatrizDispersa, [153](#)
- N
 - RectSub, [227](#)
- n
 - BandCholesky, [21](#)
 - BandSolve, [25](#)
 - CGM, [31](#)
 - DQGMRES, [63](#)
 - HeapSort, [101](#)
 - MultBandSym, [173](#)
 - SinPinxSinPiny, [234](#)
 - SinPinxSinPinySinPinz, [237](#)
- nBd
 - RectSub, [227](#)
- nD
 - CreateBdNodes, [36](#)
 - DualPrimal, [66](#)
- nDim
 - DPMainMPI, [54](#)
 - DPMethod, [59](#)
 - EllipOp, [69](#)
 - Interchange, [117](#)
 - InternalBd, [126](#)
 - RectSub, [227](#)
- nDual

- CreateBdNodes, [36](#)
 - DPMMethod, [59](#)
 - DualPrimal, [66](#)
- nFull
 - RectSub, [227](#)
- nInt
 - RectSub, [228](#)
- nIter
 - CGM, [31](#)
 - DQGMRES, [63](#)
- NMAXITER
 - Definiciones.hpp, [268](#)
- NMAXITER_LOCAL
 - Definiciones.hpp, [268](#)
- nMaxIter
 - CGM, [31](#)
 - DQGMRES, [63](#)
- nOmega
 - DPMainMPI, [54](#)
 - DPMMethod, [59](#)
 - Interchange, [118](#)
- nP
 - CreateBdNodes, [36](#)
 - DualPrimal, [66](#)
- nPrimal
 - CreateBdNodes, [36](#)
 - DualPrimal, [66](#)
- NSfExpXY, [177](#)
 - eval, [180](#)
 - getVar, [180](#)
 - NSfExpXY, [180](#)
 - NSfExpXY, [180](#)
 - setVar, [180](#)
 - var, [180](#)
- NSfExpXY.hpp, [311](#)
- NSfExpXYZ, [180](#)
 - eval, [183](#)
 - getVar, [183](#)
 - NSfExpXYZ, [183](#)
 - NSfExpXYZ, [183](#)
 - setVar, [183](#)
 - var, [183](#)
- NSfExpXYZ.hpp, [312](#)
- name
 - AllPrimal, [17](#)
 - NoPrimal, [177](#)
 - Solvable, [252](#)
 - VertEdgePrimal, [255](#)
 - VertPrimal, [258](#)
- nameClass
 - ErrorControl, [72](#)
- nameClassFunct
 - ErrorControl, [73](#)
- nameFunct
 - ErrorControl, [73](#)
- nmClass
 - ErrorControl, [73](#)
- nmFunction
 - ErrorControl, [73](#)
- Nmb
 - Matriz_Base, [147](#)
- NoPrimal, [175](#)
 - isPrimal, [177](#)
 - name, [177](#)
 - NoPrimal, [177](#)
 - NoPrimal, [177](#)
- NoPrimal.hpp, [310](#)
- node
 - BdNode, [26](#)
 - InternalBd, [126](#)
- nodeType
 - RectSub, [226](#)
- norm
 - CGM, [31](#)
- np
 - EsquemaMEMPI, [78](#)
 - RectSub, [228](#)
- npu
 - EsquemaMEMPI, [78](#)
- nta
 - DPMainMPI, [54](#)
- ntype
 - RectSub, [228](#)
- numeroProcesadoresUsar
 - EsquemaMEMPI, [77](#)
- omegas
 - DPMainMPI, [54](#)
 - Interchange, [118](#)
- one
 - DPMainMPI, [54](#)
 - Interchange, [118](#)
- op
 - DPMainMPI, [54](#)
 - Interchange, [118](#)
 - LM1, [131](#)
 - LM2, [138](#)
 - MF1, [158](#)
 - MF2, [166](#)
 - PLM1, [187](#)
 - PLM2, [194](#)
 - PMF1, [200](#)
 - PMF2, [208](#)
 - RectSub, [228](#)
- p
 - CGM, [31](#)
 - DQGMRES, [63](#)
- PLM1, [183](#)

- clean, 186
- dot, 186
- f, 187
- multOp, 186
- op, 187
- PLM1, 186
- PLM1, 186
- rhs, 186
- solve, 186
- uf, 187
- up, 187
- PLM1.cpp, 313
- PLM1.hpp, 313
- PLM1MPI, 187
 - clean, 190
 - iniInterchage, 190
 - PLM1MPI, 190
 - PLM1MPI, 190
 - solvePar, 190
- PLM1MPI.hpp, 314
- PLM2, 191
 - clean, 193
 - dot, 193
 - f, 194
 - multOp, 193
 - op, 194
 - PLM2, 193
 - PLM2, 193
 - rhs, 193
 - solve, 193
 - uf, 194
 - ug, 194
 - up, 194
- PLM2.cpp, 315
- PLM2.hpp, 315
- PLM2MPI, 194
 - clean, 197
 - iniInterchage, 197
 - PLM2MPI, 197
 - PLM2MPI, 197
 - solvePar, 197
- PLM2MPI.hpp, 316
- PMF1, 198
 - clean, 200
 - dot, 200
 - multOp, 200
 - op, 200
 - PMF1, 200
 - PMF1, 200
 - rhs, 200
 - solve, 200
- PMF1.cpp, 316
- PMF1.hpp, 317
- PMF1MPI, 201
 - clean, 204
 - iniInterchage, 204
 - PMF1MPI, 204
 - PMF1MPI, 204
 - solvePar, 204
- PMF1MPI.hpp, 318
- PMF2, 205
 - clean, 208
 - dot, 208
 - multOp, 208
 - op, 208
 - PMF2, 208
 - PMF2, 208
 - rhs, 208
 - solve, 208
 - uf, 208
 - up, 209
- PMF2.cpp, 318
- PMF2.hpp, 318
- PMF2MPI, 209
 - clean, 212
 - iniInterchage, 212
 - PMF2MPI, 212
 - PMF2MPI, 212
 - solvePar, 212
- PMF2MPI.hpp, 319
- PRIMAL
 - Primal, 215
 - RectSub, 228
- parse
 - PropDef, 218
- pbdValues
 - Interchange, 116
- prCoord
 - DPMMethod, 58
- prim
 - DPMainMPI, 54
 - Interchange, 118
- Primal, 213
 - ~Primal, 215
 - EDGE, 215
 - FACE, 215
 - INTBD, 215
 - INTERIOR, 215
 - isPrimal, 215
 - KNOWN, 215
 - PRIMAL, 215
 - Primal, 215
 - VERTEX, 215
- primal
 - DPMainMPI, 54
 - Interchange, 118
- Primal.hpp, 320
- print

- BandCholesky, [20](#)
- BandSolve, [24](#)
- DPMMethod, [58](#)
- Interchange, [116](#)
- InterchangeMPI, [123](#)
- RectSub, [226](#)
- printMat
 - RectSub, [226](#)
- printMult
 - RectSub, [226](#)
- printTime
 - DPMMethod, [58](#)
- printv
 - DPMMethod, [59](#)
- PropDef, [215](#)
 - getDouble, [218](#)
 - getInt, [218](#)
 - getString, [218](#)
 - parse, [218](#)
 - PropDef, [218](#)
 - PropDef, [218](#)
- PropDef.cpp, [320](#)
- PropDef.hpp, [321](#)
- Properties, [218](#)
 - ce, [221](#)
 - getProperty, [221](#)
 - keys, [221](#)
 - list, [221](#)
 - load, [221](#)
 - Properties, [221](#)
 - setProperty, [221](#)
 - vals, [221](#)
- Properties.cpp, [322](#)
- Properties.hpp, [322](#)
- props
 - DPMMethod, [59](#)
 - Interchange, [118](#)
- q
 - DQGMRES, [63](#)
- r
 - CGM, [31](#)
 - HeapSort, [101](#)
- RESIDUAL
 - Definiciones.hpp, [269](#)
- rbd
 - InternalBd, [126](#)
- rbdValues
 - Interchange, [116](#)
- rcoord
 - InternalBd, [126](#)
- rdp
 - InternalBd, [126](#)
- RectSub, [221](#)
- ~RectSub, [225](#)
- addProjNs, [225](#)
- bFloat, [226](#)
- bdMap, [226](#)
- bsym, [226](#)
- Ca, [226](#)
- Cb, [226](#)
- Cc, [227](#)
- ce, [227](#)
- clear, [225](#)
- coef, [227](#)
- coord, [227](#)
- coordN, [227](#)
- DUAL, [227](#)
- diff, [225](#)
- diffValues, [225](#)
- domain, [227](#)
- EDGE, [227](#)
- FACE, [227](#)
- genCoef, [225](#)
- genCoefVar, [225](#)
- genInv, [225](#)
- genInverse, [225](#)
- genNcoord, [225](#)
- genNtype, [225](#)
- getBdSize, [225](#)
- getCoord, [225](#)
- getCoordNode, [225](#)
- getInternalBd, [225](#)
- getInv, [225](#)
- getNP, [225](#)
- getNtype, [225](#)
- getPrimals, [225](#)
- getValue, [225](#)
- getValues, [225](#)
- h, [227](#)
- hfacs, [227](#)
- INTBD, [227](#)
- INTERIOR, [227](#)
- id, [227](#)
- inv, [227](#)
- inverse, [225](#)
- isDual, [225](#)
- isFloat, [226](#)
- isIntBd, [226](#)
- isInterior, [226](#)
- isKnown, [226](#)
- isPrimal, [226](#)
- isVertex, [226](#)
- KNOWN, [227](#)
- knownValues, [226](#)
- M, [227](#)
- M1, [227](#)
- mapFull, [227](#)

- mapInt, [227](#)
- mesh, [227](#)
- multOp, [226](#)
- N, [227](#)
- nBd, [227](#)
- nDim, [227](#)
- nFull, [227](#)
- nInt, [228](#)
- nodeType, [226](#)
- np, [228](#)
- ntype, [228](#)
- op, [228](#)
- PRIMAL, [228](#)
- print, [226](#)
- printMat, [226](#)
- printMult, [226](#)
- RectSub, [225](#)
- RectSub, [225](#)
- rhs, [226](#)
- scr, [228](#)
- setNtype, [226](#)
- setPrimals, [226](#)
- setValue, [226](#)
- setValues, [226](#)
- VERTEX, [228](#)
- X, [228](#)
- x, [228](#)
- Y, [228](#)
- RectSub.cpp, [323](#)
- RectSub.hpp, [324](#)
- Ren
 - Matriz_Base, [147](#)
- renglones
 - Matriz_Base, [147](#)
- reparteCargaTrabajo
 - EsquemaMEMPI, [77](#)
- retorna
 - MatrizDispersa, [153](#)
- retornaNumeroColumna
 - MatrizDispersa, [153](#)
- retornaNumeroColumnasBanda
 - MatrizDispersa, [153](#)
- retornaValorColumna
 - MatrizDispersa, [154](#)
- rhs
 - DPMMethod, [58](#)
 - Interchange, [116](#)
 - InterchangeMPI, [123](#)
 - LM1, [130](#)
 - LM2, [137](#)
 - MF1, [158](#)
 - MF2, [166](#)
 - PLM1, [186](#)
 - PLM2, [193](#)
 - PMF1, [200](#)
 - PMF2, [208](#)
 - RectSub, [226](#)
- rhss
 - DPMMethod, [59](#)
- rnode
 - InternalBd, [126](#)
- rr
 - HeapSort, [101](#)
- rsubd
 - InternalBd, [126](#)
- SP
 - DualPrimal, [66](#)
- sbdValues
 - Interchange, [116](#)
- scr
 - DPMMethod, [59](#)
 - RectSub, [228](#)
- setEpsilon
 - CGM, [31](#)
 - DQGMRES, [63](#)
- setF
 - EllipOp, [68](#)
- setG
 - EllipOp, [68](#)
- setMaxIter
 - CGM, [31](#)
 - DQGMRES, [63](#)
- setNtype
 - Interchange, [116](#)
 - InterchangeMPI, [123](#)
 - RectSub, [226](#)
- setPrimals
 - Interchange, [116](#)
 - InterchangeMPI, [124](#)
 - RectSub, [226](#)
- setProperty
 - Properties, [221](#)
- setValue
 - Interchange, [116](#)
 - RectSub, [226](#)
- setValues
 - RectSub, [226](#)
- setVar
 - Constant, [34](#)
 - Disc, [39](#)
 - Disc13, [42](#)
 - Disc14, [45](#)
 - Disc15, [48](#)
 - ExpVXY, [81](#)
 - ExpVXYZ, [84](#)
 - ExpX, [87](#)
 - ExpXY, [90](#)

- ExpXYZ, 93
- fExpXY, 96
- FunctionV1, 99
- NSfExpXY, 180
- NSfExpXYZ, 183
- SfExpXYZ, 231
- SinPinxSinPiny, 234
- SinPinxSinPinySinPinz, 237
- SinPix, 240
- SinPixCosPiy, 243
- SinPixSinPiy, 246
- SinPiXSinPiYSinPiZ, 249
- sf
 - DPMainMPI, 54
 - Interchange, 118
- SfExpXYZ, 228
 - eval, 231
 - getVar, 231
 - setVar, 231
 - SfExpXYZ, 231
 - SfExpXYZ, 231
 - var, 231
- SfExpXYZ.hpp, 325
- sg
 - DPMainMPI, 54
 - Interchange, 118
- siftup
 - HeapSort, 101
- SinPiXSinPiYSinPiZ, 246
 - eval, 249
 - getVar, 249
 - setVar, 249
 - SinPiXSinPiYSinPiZ, 249
 - SinPiXSinPiYSinPiZ, 249
 - var, 249
- SinPiXSinPiYSinPiZ.hpp, 331
- SinPinxSinPiny, 231
 - eval, 234
 - getVar, 234
 - n, 234
 - setVar, 234
 - SinPinxSinPiny, 234
 - SinPinxSinPiny, 234
 - var, 234
- SinPinxSinPiny.hpp, 326
- SinPinxSinPinySinPinz, 234
 - eval, 237
 - getVar, 237
 - n, 237
 - setVar, 237
 - SinPinxSinPinySinPinz, 237
 - SinPinxSinPinySinPinz, 237
 - var, 237
- SinPinxSinPinySinPinz.hpp, 327
- SinPix, 237
 - eval, 240
 - getVar, 240
 - setVar, 240
 - SinPix, 240
 - SinPix, 240
 - var, 240
- SinPix.hpp, 328
- SinPixCosPiy, 240
 - eval, 243
 - getVar, 243
 - setVar, 243
 - SinPixCosPiy, 243
 - SinPixCosPiy, 243
 - var, 243
- SinPixCosPiy.hpp, 329
- SinPixSinPiy, 243
 - eval, 246
 - getVar, 246
 - setVar, 246
 - SinPixSinPiy, 246
 - SinPixSinPiy, 246
 - var, 246
- SinPixSinPiy.hpp, 330
- sol
 - EllipOp, 69
- solicitaMemoria
 - MatrizDispersa, 154
- Solvable, 249
 - ~Solvable, 251
 - clean, 251
 - getIter, 252
 - getName, 252
 - name, 252
 - Solvable, 251
 - solve, 252
- Solvable.hpp, 332
- solve
 - BandCholesky, 20
 - BandSolve, 25
 - CGM, 31
 - DPMMethod, 58
 - DQGMRES, 63
 - LM1, 130
 - LM2, 137
 - MF1, 158
 - MF2, 166
 - PLM1, 186
 - PLM2, 193
 - PMF1, 200
 - PMF2, 208
 - Solvable, 252
- solveAPP
 - DualPrimal, 65

- solvePar
 - LM1MPI, [134](#)
 - LM2MPI, [141](#)
 - MF1MPI, [162](#)
 - MF2MPI, [170](#)
 - PLM1MPI, [190](#)
 - PLM2MPI, [197](#)
 - PMF1MPI, [204](#)
 - PMF2MPI, [212](#)
- solveS
 - DualPrimal, [66](#)
- solver
 - DPMethod, [59](#)
- sort
 - HeapSort, [101](#)
- subd
 - BdNode, [27](#)
 - InternalBd, [126](#)
- swap
 - HeapSort, [101](#)
- swprint
 - DPMMainMPI, [54](#)
 - Interchange, [118](#)
- t1
 - DPMMainMPI, [54](#)
- t2
 - DPMMainMPI, [54](#)
- ta
 - EsquemaMEMPI, [78](#)
- tamanoBanda
 - MatrizDispersa, [154](#)
- time0
 - DPMethod, [59](#)
- time1
 - DPMethod, [59](#)
- time2
 - DPMethod, [59](#)
- time3
 - DPMethod, [59](#)
- toSubdomains
 - DualPrimal, [66](#)
 - Interchange, [117](#)
 - InterchangeMPI, [124](#)
- u
 - DPMethod, [59](#)
- uf
 - LM1, [131](#)
 - LM2, [138](#)
 - PLM1, [187](#)
 - PLM2, [194](#)
 - PMF2, [208](#)
- ug
 - LM2, [138](#)
- PLM2, [194](#)
- up
 - LM1, [131](#)
 - LM2, [138](#)
 - MF2, [166](#)
 - PLM1, [187](#)
 - PLM2, [194](#)
 - PMF2, [209](#)
- v
 - CGM, [31](#)
 - DQGMRES, [63](#)
- VERTEX
 - Primal, [215](#)
 - RectSub, [228](#)
- val
 - ICGM, [105](#)
- vals
 - Properties, [221](#)
- var
 - Disc, [39](#)
 - Disc13, [42](#)
 - Disc14, [45](#)
 - Disc15, [48](#)
 - ExpVXY, [81](#)
 - ExpVXYZ, [84](#)
 - ExpX, [87](#)
 - ExpXY, [90](#)
 - ExpXYZ, [93](#)
 - fExpXY, [96](#)
 - NSfExpXY, [180](#)
 - NSfExpXYZ, [183](#)
 - SfExpXYZ, [231](#)
 - SinPinxSinPiny, [234](#)
 - SinPinxSinPinySinPinz, [237](#)
 - SinPix, [240](#)
 - SinPixCosPiy, [243](#)
 - SinPixSinPiy, [246](#)
 - SinPiXSinPiYSinPiZ, [249](#)
- VertEdgePrimal, [252](#)
 - isPrimal, [255](#)
 - name, [255](#)
 - VertEdgePrimal, [255](#)
 - VertEdgePrimal, [255](#)
- VertEdgePrimal.hpp, [332](#)
- VertPrimal, [255](#)
 - isPrimal, [258](#)
 - name, [258](#)
 - VertPrimal, [258](#)
 - VertPrimal, [258](#)
- VertPrimal.hpp, [333](#)
- visualiza
 - MatrizDispersa, [154](#)
- visualizaMatricesInternas

MatrizDispersa, [154](#)

X

RectSub, [228](#)

x

RectSub, [228](#)

XP

DualPrimal, [66](#)

xnp

DPMainMPI, [54](#)

InterchangeMPI, [124](#)

Y

RectSub, [228](#)

YP

DualPrimal, [66](#)

zero

DPMainMPI, [54](#)

Interchange, [118](#)