

DDM-DVS

Generated by Doxygen 1.8.9.1

Tue Sep 29 2015 08:06:05



# Contents

<b>1</b>	<b>Grupo de Modelación Matemática y Computacional, UNAM</b>	<b>1</b>
1.1	Introducción . . . . .	1
<b>2</b>	<b>Todo List</b>	<b>3</b>
<b>3</b>	<b>Bug List</b>	<b>5</b>
<b>4</b>	<b>Hierarchical Index</b>	<b>7</b>
4.1	Class Hierarchy . . . . .	7
<b>5</b>	<b>Class Index</b>	<b>9</b>
5.1	Class List . . . . .	9
<b>6</b>	<b>File Index</b>	<b>11</b>
6.1	File List . . . . .	11
<b>7</b>	<b>Class Documentation</b>	<b>15</b>
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	<a href="#">convertBand</a>	20
7.2.2.4	<a href="#">factorLU</a>	20
7.2.2.5	<a href="#">getIter</a>	20
7.2.2.6	<a href="#">print</a>	20
7.2.2.7	<a href="#">solve</a>	20
7.2.3	<a href="#">Member Data Documentation</a>	20
7.2.3.1	<a href="#">AK</a>	20
7.2.3.2	<a href="#">bw</a>	20
7.2.3.3	<a href="#">ce</a>	21
7.2.3.4	<a href="#">n</a>	21
7.3	<a href="#">BandSolve Class Reference</a>	21
7.3.1	<a href="#">Constructor &amp; Destructor Documentation</a>	24
7.3.1.1	<a href="#">BandSolve</a>	24
7.3.1.2	<a href="#">BandSolve</a>	24
7.3.1.3	<a href="#">BandSolve</a>	24
7.3.1.4	<a href="#">~BandSolve</a>	24
7.3.2	<a href="#">Member Function Documentation</a>	24
7.3.2.1	<a href="#">clean</a>	24
7.3.2.2	<a href="#">convertBand</a>	24
7.3.2.3	<a href="#">convertBand</a>	24
7.3.2.4	<a href="#">factorLU</a>	24
7.3.2.5	<a href="#">getIter</a>	24
7.3.2.6	<a href="#">print</a>	25
7.3.2.7	<a href="#">solve</a>	25
7.3.3	<a href="#">Member Data Documentation</a>	25
7.3.3.1	<a href="#">AK</a>	25
7.3.3.2	<a href="#">bw</a>	25
7.3.3.3	<a href="#">ce</a>	25
7.3.3.4	<a href="#">n</a>	25
7.4	<a href="#">BdNode Class Reference</a>	25
7.4.1	<a href="#">Constructor &amp; Destructor Documentation</a>	26
7.4.1.1	<a href="#">BdNode</a>	26
7.4.2	<a href="#">Member Data Documentation</a>	26
7.4.2.1	<a href="#">index</a>	26
7.4.2.2	<a href="#">mult</a>	26
7.4.2.3	<a href="#">node</a>	26
7.4.2.4	<a href="#">subd</a>	26

7.5	CGM Class Reference	26
7.5.1	Constructor & Destructor Documentation	29
7.5.1.1	CGM	29
7.5.1.2	CGM	29
7.5.1.3	~CGM	29
7.5.2	Member Function Documentation	29
7.5.2.1	clean	29
7.5.2.2	getIter	29
7.5.2.3	inicializa	29
7.5.2.4	norm	29
7.5.2.5	setEpsilon	29
7.5.2.6	setMaxIter	29
7.5.2.7	solve	29
7.5.3	Member Data Documentation	30
7.5.3.1	A	30
7.5.3.2	alpha	30
7.5.3.3	beta	30
7.5.3.4	ce	30
7.5.3.5	dotP	30
7.5.3.6	eps	30
7.5.3.7	gamma	30
7.5.3.8	lambda	30
7.5.3.9	mu	30
7.5.3.10	n	30
7.5.3.11	nIter	30
7.5.3.12	nMaxIter	30
7.5.3.13	p	30
7.5.3.14	r	30
7.5.3.15	v	30
7.6	Constant Class Reference	30
7.6.1	Constructor & Destructor Documentation	33
7.6.1.1	Constant	33
7.6.2	Member Function Documentation	33
7.6.2.1	eval	33
7.6.2.2	getVar	33
7.6.2.3	setVar	33
7.6.3	Member Data Documentation	33

7.6.3.1	a	33
7.7	CreateBdNodes Class Reference	33
7.7.1	Constructor & Destructor Documentation	35
7.7.1.1	CreateBdNodes	35
7.7.1.2	~CreateBdNodes	35
7.7.2	Member Data Documentation	35
7.7.2.1	bdAll	35
7.7.2.2	bdDuals	35
7.7.2.3	bdPrimals	35
7.7.2.4	ce	35
7.7.2.5	dualMult	35
7.7.2.6	hbd	35
7.7.2.7	ibd	35
7.7.2.8	ibdAll	35
7.7.2.9	maxBd	35
7.7.2.10	nD	35
7.7.2.11	nDual	35
7.7.2.12	nP	35
7.7.2.13	nPrimal	35
7.8	Disc Class Reference	35
7.8.1	Constructor & Destructor Documentation	38
7.8.1.1	Disc	38
7.8.2	Member Function Documentation	38
7.8.2.1	eval	38
7.8.2.2	getVar	38
7.8.2.3	setVar	38
7.8.3	Member Data Documentation	38
7.8.3.1	var	38
7.9	Disc13 Class Reference	38
7.9.1	Constructor & Destructor Documentation	41
7.9.1.1	Disc13	41
7.9.2	Member Function Documentation	41
7.9.2.1	eval	41
7.9.2.2	getVar	41
7.9.2.3	setVar	41
7.9.3	Member Data Documentation	41
7.9.3.1	var	41

7.10	Disc14 Class Reference	41
7.10.1	Constructor & Destructor Documentation	44
7.10.1.1	Disc14	44
7.10.2	Member Function Documentation	44
7.10.2.1	eval	44
7.10.2.2	getVar	44
7.10.2.3	setVar	44
7.10.3	Member Data Documentation	44
7.10.3.1	var	44
7.11	Disc15 Class Reference	44
7.11.1	Constructor & Destructor Documentation	47
7.11.1.1	Disc15	47
7.11.2	Member Function Documentation	47
7.11.2.1	eval	47
7.11.2.2	getVar	47
7.11.2.3	setVar	47
7.11.3	Member Data Documentation	47
7.11.3.1	var	47
7.12	DotProd Class Reference	47
7.12.1	Member Function Documentation	48
7.12.1.1	dot	48
7.13	DPMMainMPI Class Reference	49
7.13.1	Detailed Description	51
7.13.2	Constructor & Destructor Documentation	52
7.13.2.1	DPMMainMPI	52
7.13.2.2	~DPMMainMPI	52
7.13.3	Member Function Documentation	52
7.13.3.1	deleteInternalBd	52
7.13.3.2	Esclavo	52
7.13.4	Member Data Documentation	52
7.13.4.1	c	52
7.13.4.2	domain	52
7.13.4.3	f	52
7.13.4.4	fc	52
7.13.4.5	g	52
7.13.4.6	gc	52
7.13.4.7	hbd	52

7.13.4.8	indl	52
7.13.4.9	mesh	52
7.13.4.10	method	52
7.13.4.11	msa	52
7.13.4.12	mss	52
7.13.4.13	nDim	53
7.13.4.14	nOmega	53
7.13.4.15	nta	53
7.13.4.16	omegas	53
7.13.4.17	one	53
7.13.4.18	op	53
7.13.4.19	prim	53
7.13.4.20	primal	53
7.13.4.21	sf	53
7.13.4.22	sg	53
7.13.4.23	swprint	53
7.13.4.24	t1	53
7.13.4.25	t2	53
7.13.4.26	xnp	53
7.13.4.27	zero	53
7.14	DPMMethod Class Reference	53
7.14.1	Constructor & Destructor Documentation	56
7.14.1.1	DPMMethod	56
7.14.1.2	~DPMMethod	56
7.14.2	Member Function Documentation	56
7.14.2.1	analyticSolution	56
7.14.2.2	clean	56
7.14.2.3	conditionalNumber	56
7.14.2.4	genInverse	57
7.14.2.5	getSize	57
7.14.2.6	iniInterchage	57
7.14.2.7	initialize	57
7.14.2.8	prCoord	57
7.14.2.9	print	57
7.14.2.10	printTime	57
7.14.2.11	rhs	57
7.14.2.12	solve	57



7.14.3	Member Data Documentation	57
7.14.3.1	ce	57
7.14.3.2	dualp	57
7.14.3.3	epsilon	57
7.14.3.4	inter	57
7.14.3.5	nDim	57
7.14.3.6	nDual	57
7.14.3.7	nOmega	57
7.14.3.8	printv	57
7.14.3.9	props	58
7.14.3.10	rhss	58
7.14.3.11	scr	58
7.14.3.12	solver	58
7.14.3.13	time0	58
7.14.3.14	time1	58
7.14.3.15	time2	58
7.14.3.16	time3	58
7.14.3.17	u	58
7.15	DQGMRES Class Reference	58
7.15.1	Constructor & Destructor Documentation	61
7.15.1.1	DQGMRES	61
7.15.1.2	DQGMRES	61
7.15.1.3	~DQGMRES	61
7.15.2	Member Function Documentation	61
7.15.2.1	applyOmega	61
7.15.2.2	clean	61
7.15.2.3	getIter	61
7.15.2.4	inicializa	61
7.15.2.5	setEpsilon	61
7.15.2.6	setMaxIter	61
7.15.2.7	solve	61
7.15.3	Member Data Documentation	62
7.15.3.1	ce	62
7.15.3.2	cs	62
7.15.3.3	eps	62
7.15.3.4	gm	62
7.15.3.5	gm1	62

7.15.3.6	h	62
7.15.3.7	k	62
7.15.3.8	k1	62
7.15.3.9	maxIter	62
7.15.3.10	mult	62
7.15.3.11	n	62
7.15.3.12	nIter	62
7.15.3.13	nMaxIter	62
7.15.3.14	p	62
7.15.3.15	q	62
7.15.3.16	v	62
7.16	DualPrimal Class Reference	62
7.16.1	Constructor & Destructor Documentation	64
7.16.1.1	DualPrimal	64
7.16.1.2	~DualPrimal	64
7.16.2	Member Function Documentation	64
7.16.2.1	a	64
7.16.2.2	calcValues	64
7.16.2.3	fromSubdomains	64
7.16.2.4	genMats	64
7.16.2.5	getNDual	64
7.16.2.6	j	64
7.16.2.7	multS	64
7.16.2.8	solveAPP	64
7.16.2.9	solveS	64
7.16.2.10	toSubdomains	64
7.16.3	Member Data Documentation	64
7.16.3.1	ce	64
7.16.3.2	inter	64
7.16.3.3	nD	64
7.16.3.4	nDual	65
7.16.3.5	nP	65
7.16.3.6	nPrimal	65
7.16.3.7	SP	65
7.16.3.8	XP	65
7.16.3.9	YP	65
7.17	EllipOp Class Reference	65

7.17.1	Constructor & Destructor Documentation	67
7.17.1.1	EllipOp	67
7.17.1.2	EllipOp	67
7.17.1.3	~EllipOp	67
7.17.2	Member Function Documentation	67
7.17.2.1	getA	67
7.17.2.2	getB	67
7.17.2.3	getC	67
7.17.2.4	getF	67
7.17.2.5	getG	67
7.17.2.6	isSymmetric	67
7.17.2.7	setF	67
7.17.2.8	setG	67
7.17.3	Member Data Documentation	67
7.17.3.1	a	67
7.17.3.2	b	67
7.17.3.3	c	68
7.17.3.4	f	68
7.17.3.5	g	68
7.17.3.6	nDim	68
7.17.3.7	sol	68
7.18	ErrorControl Class Reference	68
7.18.1	Detailed Description	69
7.18.2	Constructor & Destructor Documentation	69
7.18.2.1	ErrorControl	69
7.18.2.2	ErrorControl	69
7.18.2.3	ErrorControl	70
7.18.3	Member Function Documentation	70
7.18.3.1	fatalError	70
7.18.3.2	fatalError	70
7.18.3.3	memoryError	70
7.18.3.4	memoryError	70
7.18.3.5	memoryError	71
7.18.3.6	nameClass	72
7.18.3.7	nameClassFunct	72
7.18.3.8	nameFunct	72
7.18.4	Member Data Documentation	72

7.18.4.1	nmClass	72
7.18.4.2	nmFunction	72
7.19	EsquemaMEMPI Class Reference	72
7.19.1	Detailed Description	75
7.19.2	Constructor & Destructor Documentation	75
7.19.2.1	EsquemaMEMPI	75
7.19.2.2	~EsquemaMEMPI	76
7.19.3	Member Function Documentation	76
7.19.3.1	generaRepartoCarga	76
7.19.3.2	numeroProcesadoresUsar	76
7.19.3.3	reparteCargaTrabajo	76
7.19.4	Member Data Documentation	76
7.19.4.1	ce	76
7.19.4.2	id	76
7.19.4.3	np	77
7.19.4.4	npu	77
7.19.4.5	ta	77
7.20	ExpVXY Class Reference	77
7.20.1	Constructor & Destructor Documentation	80
7.20.1.1	ExpVXY	80
7.20.2	Member Function Documentation	80
7.20.2.1	eval	80
7.20.2.2	getVar	80
7.20.2.3	setVar	80
7.20.3	Member Data Documentation	80
7.20.3.1	var	80
7.21	ExpVXYZ Class Reference	80
7.21.1	Constructor & Destructor Documentation	83
7.21.1.1	ExpVXYZ	83
7.21.2	Member Function Documentation	83
7.21.2.1	eval	83
7.21.2.2	getVar	83
7.21.2.3	setVar	83
7.21.3	Member Data Documentation	83
7.21.3.1	var	83
7.22	ExpX Class Reference	83
7.22.1	Constructor & Destructor Documentation	86

7.22.1.1	ExpX	86
7.22.2	Member Function Documentation	86
7.22.2.1	eval	86
7.22.2.2	getVar	86
7.22.2.3	setVar	86
7.22.3	Member Data Documentation	86
7.22.3.1	var	86
7.23	ExpXY Class Reference	86
7.23.1	Constructor & Destructor Documentation	89
7.23.1.1	ExpXY	89
7.23.2	Member Function Documentation	89
7.23.2.1	eval	89
7.23.2.2	getVar	89
7.23.2.3	setVar	89
7.23.3	Member Data Documentation	89
7.23.3.1	var	89
7.24	ExpXYZ Class Reference	89
7.24.1	Constructor & Destructor Documentation	92
7.24.1.1	ExpXYZ	92
7.24.2	Member Function Documentation	92
7.24.2.1	eval	92
7.24.2.2	getVar	92
7.24.2.3	setVar	92
7.24.3	Member Data Documentation	92
7.24.3.1	var	92
7.25	fExpXY Class Reference	92
7.25.1	Constructor & Destructor Documentation	95
7.25.1.1	fExpXY	95
7.25.2	Member Function Documentation	95
7.25.2.1	eval	95
7.25.2.2	getVar	95
7.25.2.3	setVar	95
7.25.3	Member Data Documentation	95
7.25.3.1	var	95
7.26	FunctionV Class Reference	95
7.26.1	Constructor & Destructor Documentation	96
7.26.1.1	FunctionV	96

7.26.1.2	~FunctionV	96
7.26.2	Member Function Documentation	96
7.26.2.1	dimension	96
7.26.2.2	eval	97
7.26.3	Member Data Documentation	97
7.26.3.1	dim	97
7.27	FunctionV1 Class Reference	97
7.27.1	Member Function Documentation	98
7.27.1.1	getVar	98
7.27.1.2	setVar	99
7.28	HeapSort Class Reference	99
7.28.1	Constructor & Destructor Documentation	100
7.28.1.1	HeapSort	100
7.28.1.2	~HeapSort	100
7.28.2	Member Function Documentation	100
7.28.2.1	genHeap	100
7.28.2.2	rr	100
7.28.2.3	siftup	100
7.28.2.4	sort	100
7.28.2.5	swap	100
7.28.3	Member Data Documentation	100
7.28.3.1	n	100
7.28.3.2	r	100
7.29	ICGM Class Reference	100
7.29.1	Detailed Description	103
7.29.2	Constructor & Destructor Documentation	103
7.29.2.1	ICGM	103
7.29.2.2	~ICGM	103
7.29.3	Member Function Documentation	104
7.29.3.1	clean	104
7.29.3.2	dot	104
7.29.3.3	getSize	104
7.29.3.4	multOp	104
7.29.4	Member Data Documentation	104
7.29.4.1	i	104
7.29.4.2	M	104
7.29.4.3	val	104

7.30 IDQGMRES Class Reference . . . . .	104
7.30.1 Detailed Description . . . . .	107
7.30.2 Constructor & Destructor Documentation . . . . .	107
7.30.2.1 IDQGMRES . . . . .	107
7.30.2.2 ~IDQGMRES . . . . .	107
7.30.3 Member Function Documentation . . . . .	107
7.30.3.1 clean . . . . .	107
7.30.3.2 getSize . . . . .	108
7.30.3.3 multOp . . . . .	108
7.30.4 Member Data Documentation . . . . .	108
7.30.4.1 M . . . . .	108
7.31 Interchange Class Reference . . . . .	108
7.31.1 Constructor & Destructor Documentation . . . . .	112
7.31.1.1 Interchange . . . . .	112
7.31.1.2 Interchange . . . . .	112
7.31.1.3 ~Interchange . . . . .	113
7.31.2 Member Function Documentation . . . . .	113
7.31.2.1 calcSP . . . . .	113
7.31.2.2 calcula . . . . .	113
7.31.2.3 clear . . . . .	113
7.31.2.4 clear . . . . .	113
7.31.2.5 diff . . . . .	113
7.31.2.6 diffValues . . . . .	113
7.31.2.7 fromSubdomains . . . . .	113
7.31.2.8 genGeom . . . . .	113
7.31.2.9 genInv . . . . .	113
7.31.2.10 getCoordNode . . . . .	113
7.31.2.11 getMaxBdSize . . . . .	114
7.31.2.12 getND . . . . .	114
7.31.2.13 getnDim . . . . .	114
7.31.2.14 getNDuals . . . . .	114
7.31.2.15 getnOmega . . . . .	114
7.31.2.16 getNP . . . . .	114
7.31.2.17 getNPrimals . . . . .	114
7.31.2.18 getNtype . . . . .	114
7.31.2.19 getPrimals . . . . .	114
7.31.2.20 getValue . . . . .	114

7.31.2.21	getValue	114
7.31.2.22	initialize	114
7.31.2.23	inverse	114
7.31.2.24	inverse	114
7.31.2.25	knownValues	114
7.31.2.26	multOp	115
7.31.2.27	multOp	115
7.31.2.28	pbdValues	115
7.31.2.29	print	115
7.31.2.30	print	115
7.31.2.31	rbdValues	115
7.31.2.32	rhs	115
7.31.2.33	sbdValues	115
7.31.2.34	setNtype	115
7.31.2.35	setPrimals	115
7.31.2.36	setValue	115
7.31.2.37	toSubdomains	115
7.31.3	Member Data Documentation	116
7.31.3.1	Ax	116
7.31.3.2	ax	116
7.31.3.3	Ay	116
7.31.3.4	ay	116
7.31.3.5	Az	116
7.31.3.6	az	116
7.31.3.7	bds	116
7.31.3.8	bdValues	116
7.31.3.9	bx	116
7.31.3.10	by	116
7.31.3.11	bz	116
7.31.3.12	c	116
7.31.3.13	ce	116
7.31.3.14	domain	116
7.31.3.15	f	116
7.31.3.16	fc	116
7.31.3.17	g	116
7.31.3.18	gc	116
7.31.3.19	mesh	116



7.31.3.20 method . . . . .	116
7.31.3.21 nDim . . . . .	116
7.31.3.22 nOmega . . . . .	116
7.31.3.23 omegas . . . . .	116
7.31.3.24 one . . . . .	116
7.31.3.25 op . . . . .	117
7.31.3.26 prim . . . . .	117
7.31.3.27 primal . . . . .	117
7.31.3.28 props . . . . .	117
7.31.3.29 sf . . . . .	117
7.31.3.30 sg . . . . .	117
7.31.3.31 swprint . . . . .	117
7.31.3.32 zero . . . . .	117
7.32 InterchangeMPI Class Reference . . . . .	117
7.32.1 Constructor & Destructor Documentation . . . . .	120
7.32.1.1 InterchangeMPI . . . . .	120
7.32.2 Member Function Documentation . . . . .	121
7.32.2.1 calcula . . . . .	121
7.32.2.2 clear . . . . .	121
7.32.2.3 diff . . . . .	121
7.32.2.4 diffValues . . . . .	121
7.32.2.5 fromSubdomains . . . . .	121
7.32.2.6 genInv . . . . .	121
7.32.2.7 getCoordNode . . . . .	121
7.32.2.8 getInternalBd . . . . .	121
7.32.2.9 getMaxBdSize . . . . .	121
7.32.2.10 getNtype . . . . .	121
7.32.2.11 getPrimals . . . . .	122
7.32.2.12 getValue . . . . .	122
7.32.2.13 inverse . . . . .	122
7.32.2.14 knownValues . . . . .	122
7.32.2.15 multOp . . . . .	122
7.32.2.16 print . . . . .	122
7.32.2.17 print . . . . .	122
7.32.2.18 rhs . . . . .	122
7.32.2.19 setNtype . . . . .	122
7.32.2.20 setPrimals . . . . .	122

7.32.2.21	toSubdomains	123
7.32.3	Member Data Documentation	123
7.32.3.1	indl	123
7.32.3.2	ME	123
7.32.3.3	msa	123
7.32.3.4	mss	123
7.32.3.5	xnp	123
7.33	InternalBd Class Reference	123
7.33.1	Constructor & Destructor Documentation	125
7.33.1.1	InternalBd	125
7.33.1.2	InternalBd	125
7.33.1.3	~InternalBd	125
7.33.2	Member Function Documentation	125
7.33.2.1	compareTo	125
7.33.2.2	equals	125
7.33.2.3	getval	125
7.33.2.4	rbd	125
7.33.2.5	rcoord	125
7.33.2.6	rdp	125
7.33.2.7	rnode	125
7.33.2.8	rsubd	125
7.33.3	Member Data Documentation	125
7.33.3.1	bd	125
7.33.3.2	coord	125
7.33.3.3	dp	125
7.33.3.4	nDim	125
7.33.3.5	node	125
7.33.3.6	subd	125
7.34	LM1 Class Reference	126
7.34.1	Constructor & Destructor Documentation	128
7.34.1.1	LM1	128
7.34.2	Member Function Documentation	128
7.34.2.1	clean	128
7.34.2.2	dot	128
7.34.2.3	multOp	128
7.34.2.4	rhs	128
7.34.2.5	solve	128

7.34.3	Member Data Documentation	128
7.34.3.1	f	128
7.34.3.2	op	128
7.34.3.3	uf	128
7.34.3.4	up	128
7.35	LM1MPI Class Reference	129
7.35.1	Detailed Description	132
7.35.2	Constructor & Destructor Documentation	132
7.35.2.1	LM1MPI	132
7.35.3	Member Function Documentation	132
7.35.3.1	clean	132
7.35.3.2	iniInterchage	132
7.35.3.3	solvePar	132
7.36	LM2 Class Reference	132
7.36.1	Constructor & Destructor Documentation	135
7.36.1.1	LM2	135
7.36.2	Member Function Documentation	135
7.36.2.1	clean	135
7.36.2.2	dot	135
7.36.2.3	multOp	135
7.36.2.4	rhs	135
7.36.2.5	solve	135
7.36.3	Member Data Documentation	135
7.36.3.1	f	135
7.36.3.2	op	135
7.36.3.3	uf	135
7.36.3.4	ug	136
7.36.3.5	up	136
7.37	LM2MPI Class Reference	136
7.37.1	Detailed Description	139
7.37.2	Constructor & Destructor Documentation	139
7.37.2.1	LM2MPI	139
7.37.3	Member Function Documentation	139
7.37.3.1	clean	139
7.37.3.2	iniInterchage	139
7.37.3.3	solvePar	139
7.38	LookUpFunction Class Reference	139

7.38.1	Member Function Documentation	141
7.38.1.1	getF	141
7.38.2	Member Data Documentation	141
7.38.2.1	ce	141
7.39	Matriz_Base Class Reference	141
7.39.1	Detailed Description	144
7.39.2	Constructor & Destructor Documentation	144
7.39.2.1	Matriz_Base	144
7.39.2.2	~Matriz_Base	144
7.39.3	Member Function Documentation	145
7.39.3.1	asignaNombre	145
7.39.3.2	columnas	146
7.39.3.3	renglones	146
7.39.4	Member Data Documentation	146
7.39.4.1	Ban	146
7.39.4.2	ce	146
7.39.4.3	Col	146
7.39.4.4	Nmb	146
7.39.4.5	Ren	146
7.40	MatrizDispersa Class Reference	147
7.40.1	Detailed Description	149
7.40.2	Constructor & Destructor Documentation	150
7.40.2.1	MatrizDispersa	150
7.40.2.2	MatrizDispersa	150
7.40.2.3	~MatrizDispersa	150
7.40.3	Member Function Documentation	150
7.40.3.1	asigna	150
7.40.3.2	inicializa	151
7.40.3.3	multiplica	151
7.40.3.4	retorna	151
7.40.3.5	retornaNumeroColumna	151
7.40.3.6	retornaNumeroColumnasBanda	151
7.40.3.7	retornaValorColumna	152
7.40.3.8	solicitaMemoria	152
7.40.3.9	tamanoBanda	152
7.40.3.10	visualiza	152
7.40.3.11	visualizaMatricesInternas	152

7.40.4	Member Data Documentation	153
7.40.4.1	J	153
7.40.4.2	M	153
7.41	MF1 Class Reference	153
7.41.1	Constructor & Destructor Documentation	156
7.41.1.1	MF1	156
7.41.2	Member Function Documentation	156
7.41.2.1	clean	156
7.41.2.2	dot	156
7.41.2.3	multOp	156
7.41.2.4	rhs	156
7.41.2.5	solve	156
7.41.3	Member Data Documentation	156
7.41.3.1	op	156
7.42	MF1MPI Class Reference	156
7.42.1	Detailed Description	159
7.42.2	Constructor & Destructor Documentation	159
7.42.2.1	MF1MPI	159
7.42.3	Member Function Documentation	159
7.42.3.1	clean	159
7.42.3.2	iniInterchage	159
7.42.3.3	solvePar	159
7.43	MF2 Class Reference	159
7.43.1	Constructor & Destructor Documentation	162
7.43.1.1	MF2	162
7.43.2	Member Function Documentation	162
7.43.2.1	clean	162
7.43.2.2	dot	162
7.43.2.3	multOp	162
7.43.2.4	rhs	162
7.43.2.5	solve	162
7.43.3	Member Data Documentation	162
7.43.3.1	op	162
7.43.3.2	up	162
7.44	MF2MPI Class Reference	163
7.44.1	Detailed Description	165
7.44.2	Constructor & Destructor Documentation	165

7.44.2.1	MF2MPI	165
7.44.3	Member Function Documentation	165
7.44.3.1	clean	165
7.44.3.2	iniInterchage	165
7.44.3.3	solvePar	165
7.45	MultBandSym Class Reference	165
7.45.1	Constructor & Destructor Documentation	167
7.45.1.1	MultBandSym	167
7.45.2	Member Function Documentation	167
7.45.2.1	getSize	167
7.45.2.2	multOp	168
7.45.3	Member Data Documentation	168
7.45.3.1	AK	168
7.45.3.2	bw	168
7.45.3.3	n	168
7.46	MultOp Class Reference	168
7.46.1	Member Function Documentation	169
7.46.1.1	getSize	169
7.46.1.2	multOp	169
7.47	NoPrimal Class Reference	169
7.47.1	Constructor & Destructor Documentation	172
7.47.1.1	NoPrimal	172
7.47.2	Member Function Documentation	172
7.47.2.1	isPrimal	172
7.47.3	Member Data Documentation	172
7.47.3.1	name	172
7.48	NSfExpXY Class Reference	172
7.48.1	Constructor & Destructor Documentation	175
7.48.1.1	NSfExpXY	175
7.48.2	Member Function Documentation	175
7.48.2.1	eval	175
7.48.2.2	getVar	175
7.48.2.3	setVar	175
7.48.3	Member Data Documentation	175
7.48.3.1	var	175
7.49	NSfExpXYZ Class Reference	175
7.49.1	Constructor & Destructor Documentation	178

7.49.1.1	NSfExpXYZ	178
7.49.2	Member Function Documentation	178
7.49.2.1	eval	178
7.49.2.2	getVar	178
7.49.2.3	setVar	178
7.49.3	Member Data Documentation	178
7.49.3.1	var	178
7.50	PLM1 Class Reference	178
7.50.1	Constructor & Destructor Documentation	181
7.50.1.1	PLM1	181
7.50.2	Member Function Documentation	181
7.50.2.1	clean	181
7.50.2.2	dot	181
7.50.2.3	multOp	181
7.50.2.4	rhs	181
7.50.2.5	solve	181
7.50.3	Member Data Documentation	181
7.50.3.1	f	181
7.50.3.2	op	181
7.50.3.3	uf	181
7.50.3.4	up	181
7.51	PLM1MPI Class Reference	182
7.51.1	Detailed Description	185
7.51.2	Constructor & Destructor Documentation	185
7.51.2.1	PLM1MPI	185
7.51.3	Member Function Documentation	185
7.51.3.1	clean	185
7.51.3.2	iniInterchage	185
7.51.3.3	solvePar	185
7.52	PLM2 Class Reference	185
7.52.1	Constructor & Destructor Documentation	188
7.52.1.1	PLM2	188
7.52.2	Member Function Documentation	188
7.52.2.1	clean	188
7.52.2.2	dot	188
7.52.2.3	multOp	188
7.52.2.4	rhs	188

7.52.2.5	<a href="#">solve</a>	188
7.52.3	<a href="#">Member Data Documentation</a>	188
7.52.3.1	<a href="#">f</a>	188
7.52.3.2	<a href="#">op</a>	188
7.52.3.3	<a href="#">uf</a>	188
7.52.3.4	<a href="#">ug</a>	189
7.52.3.5	<a href="#">up</a>	189
7.53	<a href="#">PLM2MPI Class Reference</a>	189
7.53.1	<a href="#">Detailed Description</a>	192
7.53.2	<a href="#">Constructor &amp; Destructor Documentation</a>	192
7.53.2.1	<a href="#">PLM2MPI</a>	192
7.53.3	<a href="#">Member Function Documentation</a>	192
7.53.3.1	<a href="#">clean</a>	192
7.53.3.2	<a href="#">iniInterchage</a>	192
7.53.3.3	<a href="#">solvePar</a>	192
7.54	<a href="#">PMF1 Class Reference</a>	192
7.54.1	<a href="#">Constructor &amp; Destructor Documentation</a>	195
7.54.1.1	<a href="#">PMF1</a>	195
7.54.2	<a href="#">Member Function Documentation</a>	195
7.54.2.1	<a href="#">clean</a>	195
7.54.2.2	<a href="#">dot</a>	195
7.54.2.3	<a href="#">multOp</a>	195
7.54.2.4	<a href="#">rhs</a>	195
7.54.2.5	<a href="#">solve</a>	195
7.54.3	<a href="#">Member Data Documentation</a>	195
7.54.3.1	<a href="#">op</a>	195
7.55	<a href="#">PMF1MPI Class Reference</a>	195
7.55.1	<a href="#">Detailed Description</a>	198
7.55.2	<a href="#">Constructor &amp; Destructor Documentation</a>	198
7.55.2.1	<a href="#">PMF1MPI</a>	198
7.55.3	<a href="#">Member Function Documentation</a>	198
7.55.3.1	<a href="#">clean</a>	198
7.55.3.2	<a href="#">iniInterchage</a>	198
7.55.3.3	<a href="#">solvePar</a>	198
7.56	<a href="#">PMF2 Class Reference</a>	198
7.56.1	<a href="#">Constructor &amp; Destructor Documentation</a>	201
7.56.1.1	<a href="#">PMF2</a>	201



7.56.2	Member Function Documentation	201
7.56.2.1	clean	201
7.56.2.2	dot	201
7.56.2.3	multOp	201
7.56.2.4	rhs	201
7.56.2.5	solve	201
7.56.3	Member Data Documentation	201
7.56.3.1	op	201
7.56.3.2	uf	201
7.56.3.3	up	201
7.57	PMF2MPI Class Reference	202
7.57.1	Detailed Description	204
7.57.2	Constructor & Destructor Documentation	204
7.57.2.1	PMF2MPI	204
7.57.3	Member Function Documentation	204
7.57.3.1	clean	204
7.57.3.2	iniInterchage	204
7.57.3.3	solvePar	204
7.58	Primal Class Reference	204
7.58.1	Constructor & Destructor Documentation	206
7.58.1.1	Primal	206
7.58.1.2	~Primal	207
7.58.2	Member Function Documentation	207
7.58.2.1	isPrimal	207
7.58.3	Member Data Documentation	207
7.58.3.1	EDGE	207
7.58.3.2	FACE	207
7.58.3.3	INTBD	207
7.58.3.4	INTERIOR	207
7.58.3.5	KNOWN	207
7.58.3.6	PRIMAL	207
7.58.3.7	VERTEX	207
7.59	PropDef Class Reference	207
7.59.1	Constructor & Destructor Documentation	210
7.59.1.1	PropDef	210
7.59.1.2	PropDef	210
7.59.1.3	PropDef	210

7.59.2	Member Function Documentation	210
7.59.2.1	getDouble	210
7.59.2.2	getDouble	210
7.59.2.3	getInt	210
7.59.2.4	getInt	210
7.59.2.5	getString	210
7.59.2.6	getString	210
7.59.2.7	parse	210
7.59.2.8	parse	210
7.60	Properties Class Reference	210
7.60.1	Constructor & Destructor Documentation	213
7.60.1.1	Properties	213
7.60.2	Member Function Documentation	213
7.60.2.1	getProperty	213
7.60.2.2	getProperty	213
7.60.2.3	list	213
7.60.2.4	load	213
7.60.2.5	setProperty	213
7.60.3	Member Data Documentation	213
7.60.3.1	ce	213
7.60.3.2	keys	213
7.60.3.3	vals	213
7.61	RectSub Class Reference	213
7.61.1	Constructor & Destructor Documentation	216
7.61.1.1	RectSub	216
7.61.1.2	~RectSub	216
7.61.2	Member Function Documentation	216
7.61.2.1	clear	216
7.61.2.2	diff	217
7.61.2.3	diffValues	217
7.61.2.4	genCoef	217
7.61.2.5	genCoefVar	217
7.61.2.6	genInv	217
7.61.2.7	genInverse	217
7.61.2.8	genNcoord	217
7.61.2.9	genNtype	217
7.61.2.10	getBdSize	217

7.61.2.11	getCoord	217
7.61.2.12	getCoordNode	217
7.61.2.13	getInternalBd	217
7.61.2.14	getInv	217
7.61.2.15	getNP	217
7.61.2.16	getNtype	217
7.61.2.17	getPrimals	217
7.61.2.18	getValue	217
7.61.2.19	getValues	217
7.61.2.20	inverse	217
7.61.2.21	isDual	217
7.61.2.22	isFloat	217
7.61.2.23	isIntBd	217
7.61.2.24	isInterior	217
7.61.2.25	isInterior	217
7.61.2.26	isKnown	217
7.61.2.27	isKnown	217
7.61.2.28	isPrimal	218
7.61.2.29	isVertex	218
7.61.2.30	knownValues	218
7.61.2.31	multOp	218
7.61.2.32	nodeType	218
7.61.2.33	print	218
7.61.2.34	print	218
7.61.2.35	printMat	218
7.61.2.36	printMult	218
7.61.2.37	rhs	218
7.61.2.38	setNtype	218
7.61.2.39	setPrimals	218
7.61.2.40	setValue	218
7.61.2.41	setValues	218
7.61.3	Member Data Documentation	218
7.61.3.1	bdMap	218
7.61.3.2	bFloat	218
7.61.3.3	bsym	218
7.61.3.4	Ca	218
7.61.3.5	Cb	218

7.61.3.6 Cc	218
7.61.3.7 ce	218
7.61.3.8 coef	218
7.61.3.9 coord	218
7.61.3.10 coordN	218
7.61.3.11 domain	219
7.61.3.12 DUAL	219
7.61.3.13 EDGE	219
7.61.3.14 FACE	219
7.61.3.15 h	219
7.61.3.16 hfac	219
7.61.3.17 id	219
7.61.3.18 INTBD	219
7.61.3.19 INTERIOR	219
7.61.3.20 inv	219
7.61.3.21 KNOWN	219
7.61.3.22 M	219
7.61.3.23 M1	219
7.61.3.24 mapFull	219
7.61.3.25 mapInt	219
7.61.3.26 mesh	219
7.61.3.27 N	219
7.61.3.28 nBd	219
7.61.3.29 nDim	219
7.61.3.30 nFull	219
7.61.3.31 nInt	219
7.61.3.32 np	219
7.61.3.33 ntype	219
7.61.3.34 op	219
7.61.3.35 PRIMAL	219
7.61.3.36 scr	219
7.61.3.37 VERTEX	220
7.61.3.38 X	220
7.61.3.39 x	220
7.61.3.40 Y	220
7.62 SfExpXYZ Class Reference	220
7.62.1 Constructor & Destructor Documentation	223

7.62.1.1	SfExpXYZ	223
7.62.2	Member Function Documentation	223
7.62.2.1	eval	223
7.62.2.2	getVar	223
7.62.2.3	setVar	223
7.62.3	Member Data Documentation	223
7.62.3.1	var	223
7.63	SinPinxSinPiny Class Reference	223
7.63.1	Constructor & Destructor Documentation	226
7.63.1.1	SinPinxSinPiny	226
7.63.2	Member Function Documentation	226
7.63.2.1	eval	226
7.63.2.2	getVar	226
7.63.2.3	setVar	226
7.63.3	Member Data Documentation	226
7.63.3.1	n	226
7.63.3.2	var	226
7.64	SinPinxSinPinySinPinz Class Reference	226
7.64.1	Constructor & Destructor Documentation	229
7.64.1.1	SinPinxSinPinySinPinz	229
7.64.2	Member Function Documentation	229
7.64.2.1	eval	229
7.64.2.2	getVar	229
7.64.2.3	setVar	229
7.64.3	Member Data Documentation	229
7.64.3.1	n	229
7.64.3.2	var	229
7.65	SinPix Class Reference	229
7.65.1	Constructor & Destructor Documentation	232
7.65.1.1	SinPix	232
7.65.2	Member Function Documentation	232
7.65.2.1	eval	232
7.65.2.2	getVar	232
7.65.2.3	setVar	232
7.65.3	Member Data Documentation	232
7.65.3.1	var	232
7.66	SinPixCosPiy Class Reference	232

7.66.1	Constructor & Destructor Documentation	235
7.66.1.1	SinPixCosPiy	235
7.66.2	Member Function Documentation	235
7.66.2.1	eval	235
7.66.2.2	getVar	235
7.66.2.3	setVar	235
7.66.3	Member Data Documentation	235
7.66.3.1	var	235
7.67	SinPixSinPiy Class Reference	235
7.67.1	Constructor & Destructor Documentation	238
7.67.1.1	SinPixSinPiy	238
7.67.2	Member Function Documentation	238
7.67.2.1	eval	238
7.67.2.2	getVar	238
7.67.2.3	setVar	238
7.67.3	Member Data Documentation	238
7.67.3.1	var	238
7.68	SinPiXSinPiYSinPiZ Class Reference	238
7.68.1	Constructor & Destructor Documentation	241
7.68.1.1	SinPiXSinPiYSinPiZ	241
7.68.2	Member Function Documentation	241
7.68.2.1	eval	241
7.68.2.2	getVar	241
7.68.2.3	setVar	241
7.68.3	Member Data Documentation	241
7.68.3.1	var	241
7.69	Solvable Class Reference	241
7.69.1	Constructor & Destructor Documentation	243
7.69.1.1	Solvable	243
7.69.1.2	~Solvable	243
7.69.2	Member Function Documentation	243
7.69.2.1	clean	244
7.69.2.2	getIter	244
7.69.2.3	getName	244
7.69.2.4	solve	244
7.69.3	Member Data Documentation	244
7.69.3.1	name	244

7.70 VertEdgePrimal Class Reference . . . . .	244
7.70.1 Constructor & Destructor Documentation . . . . .	247
7.70.1.1 VertEdgePrimal . . . . .	247
7.70.2 Member Function Documentation . . . . .	247
7.70.2.1 isPrimal . . . . .	247
7.70.3 Member Data Documentation . . . . .	247
7.70.3.1 name . . . . .	247
7.71 VertPrimal Class Reference . . . . .	247
7.71.1 Constructor & Destructor Documentation . . . . .	250
7.71.1.1 VertPrimal . . . . .	250
7.71.2 Member Function Documentation . . . . .	250
7.71.2.1 isPrimal . . . . .	250
7.71.3 Member Data Documentation . . . . .	250
7.71.3.1 name . . . . .	250
<b>8 File Documentation</b>	<b>251</b>
8.1 AllPrimal.hpp File Reference . . . . .	251
8.2 BandCholesky.cpp File Reference . . . . .	252
8.3 BandCholesky.hpp File Reference . . . . .	252
8.4 BandSolve.cpp File Reference . . . . .	254
8.5 BandSolve.hpp File Reference . . . . .	254
8.6 BdNode.hpp File Reference . . . . .	256
8.7 CGM.cpp File Reference . . . . .	256
8.8 CGM.hpp File Reference . . . . .	256
8.9 Constant.hpp File Reference . . . . .	257
8.10 CreateBdNodes.hpp File Reference . . . . .	258
8.11 Definiciones.hpp File Reference . . . . .	259
8.11.1 Macro Definition Documentation . . . . .	260
8.11.1.1 __Double__ . . . . .	260
8.11.1.2 COEFICIENTES_CONSTANTES . . . . .	260
8.11.1.3 DIM_VECTOR . . . . .	260
8.11.1.4 EPS_EQUAL . . . . .	260
8.11.1.5 EPSILON . . . . .	260
8.11.1.6 EPSILON_LOCAL . . . . .	260
8.11.1.7 NMAXITER . . . . .	260
8.11.1.8 NMAXITER_LOCAL . . . . .	260
8.11.2 Typedef Documentation . . . . .	260

8.11.2.1 Idouble . . . . .	260
8.12 Disc.hpp File Reference . . . . .	261
8.13 Disc13.hpp File Reference . . . . .	262
8.14 Disc14.hpp File Reference . . . . .	263
8.15 Disc15.hpp File Reference . . . . .	264
8.16 DotProd.hpp File Reference . . . . .	265
8.17 DPMainMPI.cpp File Reference . . . . .	265
8.18 DPMainMPI.hpp File Reference . . . . .	266
8.19 DPMethod.cpp File Reference . . . . .	266
8.20 DPMethod.hpp File Reference . . . . .	267
8.21 DQGMRES.cpp File Reference . . . . .	268
8.22 DQGMRES.hpp File Reference . . . . .	268
8.23 DualPrimal.cpp File Reference . . . . .	269
8.24 DualPrimal.hpp File Reference . . . . .	269
8.25 EllipOp.hpp File Reference . . . . .	270
8.26 ErrorControl.cpp File Reference . . . . .	271
8.27 ErrorControl.hpp File Reference . . . . .	271
8.28 EsquemaMEMPI.cpp File Reference . . . . .	272
8.29 EsquemaMEMPI.hpp File Reference . . . . .	273
8.30 ExpVXY.hpp File Reference . . . . .	273
8.31 ExpVXYZ.hpp File Reference . . . . .	274
8.32 ExpX.hpp File Reference . . . . .	276
8.33 ExpXY.hpp File Reference . . . . .	277
8.34 ExpXYZ.hpp File Reference . . . . .	278
8.35 fExpXY.hpp File Reference . . . . .	279
8.36 FunctionV.hpp File Reference . . . . .	280
8.37 FunctionV1.hpp File Reference . . . . .	280
8.38 HeapSort.hpp File Reference . . . . .	281
8.39 ICGM.hpp File Reference . . . . .	282
8.40 IDQGMRES.hpp File Reference . . . . .	283
8.41 Interchange.cpp File Reference . . . . .	284
8.42 Interchange.hpp File Reference . . . . .	285
8.43 InterchangeMPI.cpp File Reference . . . . .	286
8.44 InterchangeMPI.hpp File Reference . . . . .	286
8.45 InternalBd.hpp File Reference . . . . .	287
8.46 LM1.cpp File Reference . . . . .	287
8.47 LM1.hpp File Reference . . . . .	288



8.48 LM1MPI.hpp File Reference . . . . .	289
8.49 LM2.cpp File Reference . . . . .	289
8.50 LM2.hpp File Reference . . . . .	289
8.51 LM2MPI.hpp File Reference . . . . .	290
8.52 LookUpFunction.cpp File Reference . . . . .	291
8.53 LookUpFunction.hpp File Reference . . . . .	291
8.54 main.hpp File Reference . . . . .	292
8.55 Matriz_Base.hpp File Reference . . . . .	292
8.56 MatrizDispersa.cpp File Reference . . . . .	293
8.57 MatrizDispersa.hpp File Reference . . . . .	294
8.58 MF1.cpp File Reference . . . . .	295
8.59 MF1.hpp File Reference . . . . .	296
8.60 MF1MPI.hpp File Reference . . . . .	297
8.61 MF2.cpp File Reference . . . . .	297
8.62 MF2.hpp File Reference . . . . .	297
8.63 MF2MPI.hpp File Reference . . . . .	298
8.64 MultBandSym.cpp File Reference . . . . .	299
8.65 MultBandSym.hpp File Reference . . . . .	299
8.66 MultOp.hpp File Reference . . . . .	300
8.67 NoPrimal.hpp File Reference . . . . .	301
8.68 NSfExpXY.hpp File Reference . . . . .	302
8.69 NSfExpXYZ.hpp File Reference . . . . .	303
8.70 PLM1.cpp File Reference . . . . .	304
8.71 PLM1.hpp File Reference . . . . .	304
8.72 PLM1MPI.hpp File Reference . . . . .	305
8.73 PLM2.cpp File Reference . . . . .	306
8.74 PLM2.hpp File Reference . . . . .	306
8.75 PLM2MPI.hpp File Reference . . . . .	307
8.76 PMF1.cpp File Reference . . . . .	307
8.77 PMF1.hpp File Reference . . . . .	308
8.78 PMF1MPI.hpp File Reference . . . . .	309
8.79 PMF2.cpp File Reference . . . . .	309
8.80 PMF2.hpp File Reference . . . . .	309
8.81 PMF2MPI.hpp File Reference . . . . .	310
8.82 Primal.hpp File Reference . . . . .	311
8.83 PropDef.cpp File Reference . . . . .	311
8.84 PropDef.hpp File Reference . . . . .	312

8.85	Properties.cpp File Reference . . . . .	312
8.86	Properties.hpp File Reference . . . . .	313
8.87	RectSub.cpp File Reference . . . . .	314
8.88	RectSub.hpp File Reference . . . . .	314
8.89	SfExpXYZ.hpp File Reference . . . . .	315
8.90	SinPinxSinPiny.hpp File Reference . . . . .	316
8.91	SinPinxSinPinySinPinz.hpp File Reference . . . . .	318
8.92	SinPix.hpp File Reference . . . . .	319
8.93	SinPixCosPiy.hpp File Reference . . . . .	320
8.94	SinPixSinPiy.hpp File Reference . . . . .	321
8.95	SinPiXSinPiYSinPiZ.hpp File Reference . . . . .	322
8.96	Solvable.hpp File Reference . . . . .	323
8.97	VertEdgePrimal.hpp File Reference . . . . .	323
8.98	VertPrimal.hpp File Reference . . . . .	324
<b>9</b>	<b>Example Documentation</b>	<b>327</b>
9.1	EjemploMatrizDispersa.cpp . . . . .	327
9.2	ExampleErrorControl.cpp . . . . .	327
	<b>Index</b>	<b>329</b>

## 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

# Hierarchical Index

### 4.1 Class Hierarchy

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

BdNode . . . . .	25
CreateBdNodes . . . . .	33
DotProd . . . . .	47
DPMMethod . . . . .	53
LM1 . . . . .	126
LM1MPI . . . . .	129
LM2 . . . . .	132
LM2MPI . . . . .	136
MF1 . . . . .	153
MF1MPI . . . . .	156
MF2 . . . . .	159
MF2MPI . . . . .	163
PLM1 . . . . .	178
PLM1MPI . . . . .	182
PLM2 . . . . .	185
PLM2MPI . . . . .	189
PMF1 . . . . .	192
PMF1MPI . . . . .	195
PMF2 . . . . .	198
PMF2MPI . . . . .	202
ICGM . . . . .	100
DualPrimal . . . . .	62
EllipOp . . . . .	65
ErrorControl . . . . .	68
EsquemaMEMPI . . . . .	72
DPMMainMPI . . . . .	49
LM1MPI . . . . .	129
LM2MPI . . . . .	136
MF1MPI . . . . .	156
MF2MPI . . . . .	163
PLM1MPI . . . . .	182
PLM2MPI . . . . .	189

PMF1MPI . . . . .	195
PMF2MPI . . . . .	202
FunctionV . . . . .	95
FunctionV1 . . . . .	97
Constant . . . . .	30
Disc . . . . .	35
Disc13 . . . . .	38
Disc14 . . . . .	41
Disc15 . . . . .	44
ExpVXY . . . . .	77
ExpVXYZ . . . . .	80
ExpX . . . . .	83
ExpXY . . . . .	86
ExpXYZ . . . . .	89
fExpXY . . . . .	92
NSfExpXY . . . . .	172
NSfExpXYZ . . . . .	175
SfExpXYZ . . . . .	220
SinPinxSinPiny . . . . .	223
SinPinxSinPinySinPinz . . . . .	226
SinPix . . . . .	229
SinPixCosPiy . . . . .	232
SinPixSinPiy . . . . .	235
SinPiXSinPiYSinPiZ . . . . .	238
HeapSort . . . . .	99
Interchange . . . . .	108
InterchangeMPI . . . . .	117
InternalBd . . . . .	123
LookUpFunction . . . . .	139
Matriz_Base . . . . .	141
MatrizDispersa . . . . .	147
MultOp . . . . .	168
DPMethod . . . . .	53
ICGM . . . . .	100
IDQGMRES . . . . .	104
MultBandSym . . . . .	165
Primal . . . . .	204
AllPrimal . . . . .	15
NoPrimal . . . . .	169
VertEdgePrimal . . . . .	244
VertPrimal . . . . .	247
Properties . . . . .	210
PropDef . . . . .	207
RectSub . . . . .	213
Solvable . . . . .	241
BandCholesky . . . . .	17
BandSolve . . . . .	21
CGM . . . . .	26
ICGM . . . . .	100
DQGMRES . . . . .	58
IDQGMRES . . . . .	104

## 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	26
Constant	30
CreateBdNodes	33
Disc	35
Disc13	38
Disc14	41
Disc15	44
DotProd	47
DPMMainMPI	
Clase base para definir a los metodos DVS-DDM	49
DPMMethod	53
DQGMRES	58
DualPrimal	62
EllipOp	65
ErrorControl	
Error Control	68
EsquemaMEMPI	
Clase base para definir el Esquema Maestro-Esclavo en MPI	72
ExpVXY	77
ExpVXYZ	80
ExpX	83
ExpXY	86
ExpXYZ	89
fExpXY	92
FunctionV	95
FunctionV1	97
HeapSort	99
ICGM	
Clase para implementar <a href="#">CGM</a> con matrices bandadas o dispersas	100

IDQGMRES	
Clase para implementar <a href="#">DQGMRES</a> con matrices bandadas o dispersas	104
Interchange	108
InterchangeMPI	117
InternalBd	123
LM1	126
LM1MPI	
Clase para definir el metodo LM-1 de DVS-DDM	129
LM2	132
LM2MPI	
Clase para definir el metodo LM-2 de DVS-DDM	136
LookUpFunction	139
Matriz_Base	
Clase base para el trabajar con matrices	141
MatrizDispersa	147
MF1	153
MF1MPI	
Clase para definir el metodo MF-1 de DVS-DDM	156
MF2	159
MF2MPI	
Clase para definir el metodo MF-2 de DVS-DDM	163
MultBandSym	165
MultOp	168
NoPrimal	169
NSfExpXY	172
NSfExpXYZ	175
PLM1	178
PLM1MPI	
Clase para definir el metodo PLM-1 de DVS-DDM	182
PLM2	185
PLM2MPI	
Clase para definir el metodo MF-1 de DVS-DDM	189
PMF1	192
PMF1MPI	
Clase para definir el metodo PMF-1 de DVS-DDM	195
PMF2	198
PMF2MPI	
Clase para definir el metodo PMF-2 de DVS-DDM	202
Primal	204
PropDef	207
Properties	210
RectSub	213
SfExpXYZ	220
SinPinxSinPiny	223
SinPinxSinPinySinPinz	226
SinPix	229
SinPixCosPiy	232
SinPixSinPiy	235
SinPiXSinPiYSinPiZ	238
Solvable	241
VertEdgePrimal	244
VertPrimal	247

## Chapter 6

# File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

AllPrimal.hpp	251
BandCholesky.cpp	252
BandCholesky.hpp	252
BandSolve.cpp	254
BandSolve.hpp	254
BdNode.hpp	256
CGM.cpp	256
CGM.hpp	256
Constant.hpp	257
CreateBdNodes.hpp	258
Definiciones.hpp	259
Disc.hpp	261
Disc13.hpp	262
Disc14.hpp	263
Disc15.hpp	264
DotProd.hpp	265
DPMMainMPI.cpp	265
DPMMainMPI.hpp	266
DPMMethod.cpp	266
DPMMethod.hpp	267
DQGMRES.cpp	268
DQGMRES.hpp	268
DualPrimal.cpp	269
DualPrimal.hpp	269
EllipOp.hpp	270
ErrorControl.cpp	271
ErrorControl.hpp	271
EsquemaMEMPI.cpp	272
EsquemaMEMPI.hpp	273
ExpVXY.hpp	273
ExpVXYZ.hpp	274
ExpX.hpp	276
ExpXY.hpp	277
ExpXYZ.hpp	278

fExpXY.hpp	279
FunctionV.hpp	280
FunctionV1.hpp	280
HeapSort.hpp	281
ICGM.hpp	282
IDQGMRES.hpp	283
Interchange.cpp	284
Interchange.hpp	285
InterchangeMPI.cpp	286
InterchangeMPI.hpp	286
InternalBd.hpp	287
LM1.cpp	287
LM1.hpp	288
LM1MPI.hpp	289
LM2.cpp	289
LM2.hpp	289
LM2MPI.hpp	290
LookUpFunction.cpp	291
LookUpFunction.hpp	291
main.hpp	292
Matriz_Base.hpp	292
MatrizDispersa.cpp	293
MatrizDispersa.hpp	294
MF1.cpp	295
MF1.hpp	296
MF1MPI.hpp	297
MF2.cpp	297
MF2.hpp	297
MF2MPI.hpp	298
MultBandSym.cpp	299
MultBandSym.hpp	299
MultOp.hpp	300
NoPrimal.hpp	301
NSfExpXY.hpp	302
NSfExpXYZ.hpp	303
PLM1.cpp	304
PLM1.hpp	304
PLM1MPI.hpp	305
PLM2.cpp	306
PLM2.hpp	306
PLM2MPI.hpp	307
PMF1.cpp	307
PMF1.hpp	308
PMF1MPI.hpp	309
PMF2.cpp	309
PMF2.hpp	309
PMF2MPI.hpp	310
Primal.hpp	311
PropDef.cpp	311
PropDef.hpp	312
Properties.cpp	312
Properties.hpp	313
RectSub.cpp	314
RectSub.hpp	314

SfExpXYZ.hpp	315
SinPinxSinPiny.hpp	316
SinPinxSinPinySinPinz.hpp	318
SinPix.hpp	319
SinPixCosPiy.hpp	320
SinPixSinPiy.hpp	321
SinPiXSinPiYSinPiZ.hpp	322
Solvable.hpp	323
VertEdgePrimal.hpp	323
VertPrimal.hpp	324





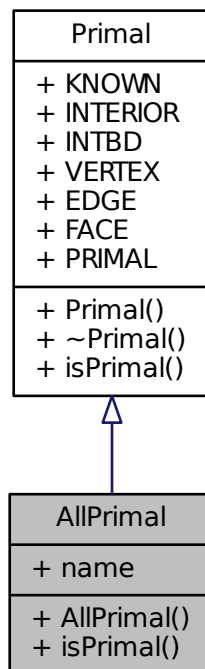
## Chapter 7

# Class Documentation

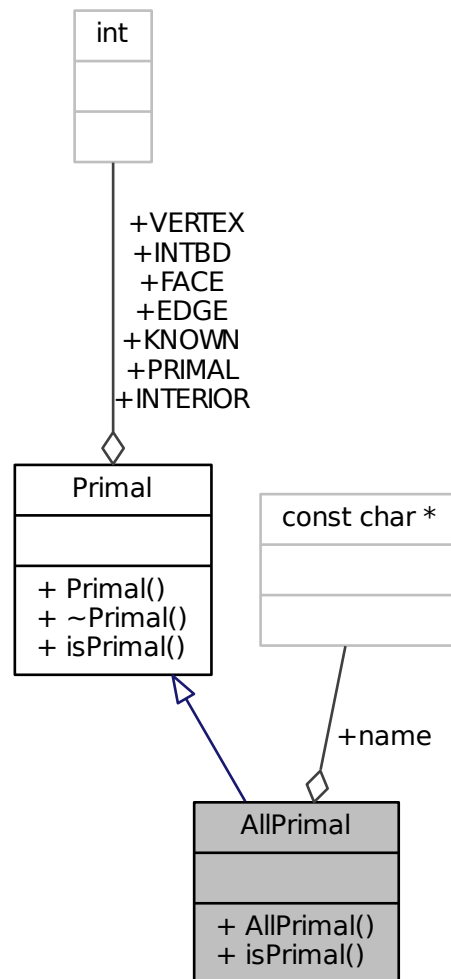
### 7.1 AllPrimal Class Reference

```
#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`

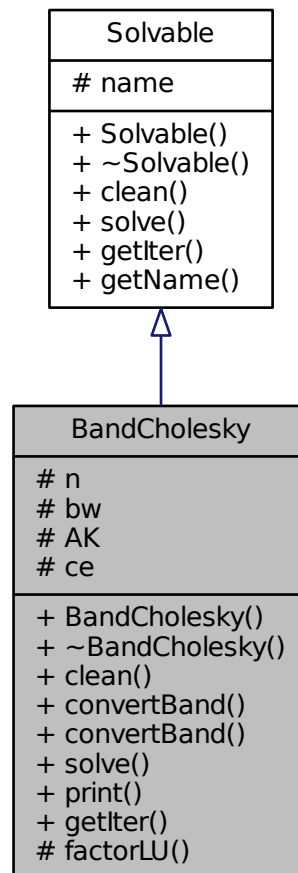
The documentation for this class was generated from the following file:

- [AllPrimal.hpp](#)

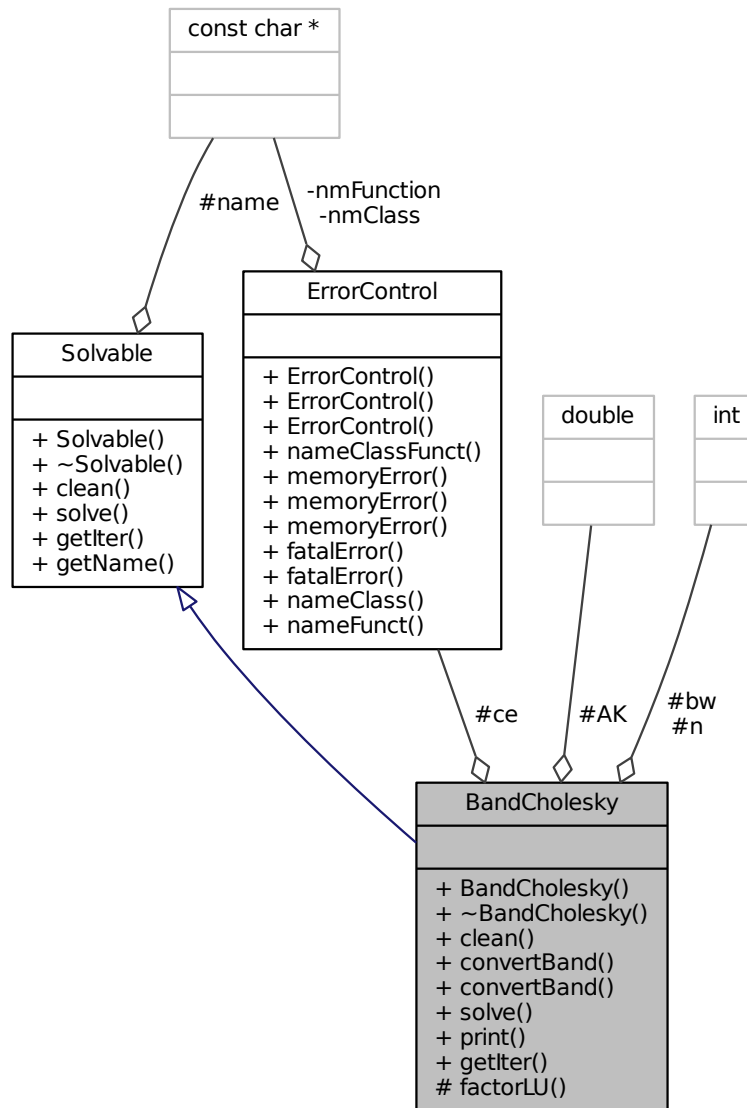
## 7.2 BandCholesky Class Reference

```
#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, Idouble \*\*A)
- void `convertBand` (int n, MatrizDispersa \*A)
- void `solve` (Idouble \*x, Idouble \*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, Idouble ** 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 ( Idouble * x, Idouble * 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]

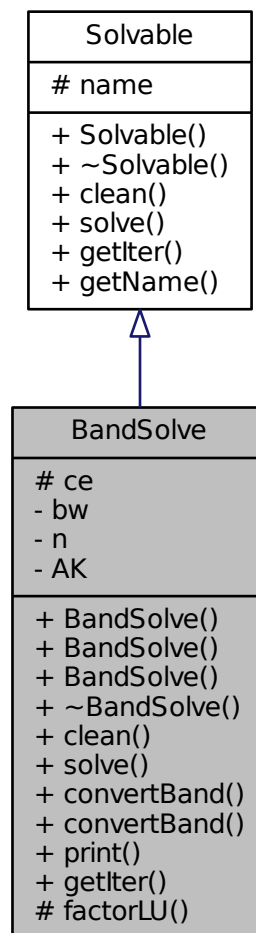
The documentation for this class was generated from the following files:

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

## 7.3 BandSolve Class Reference

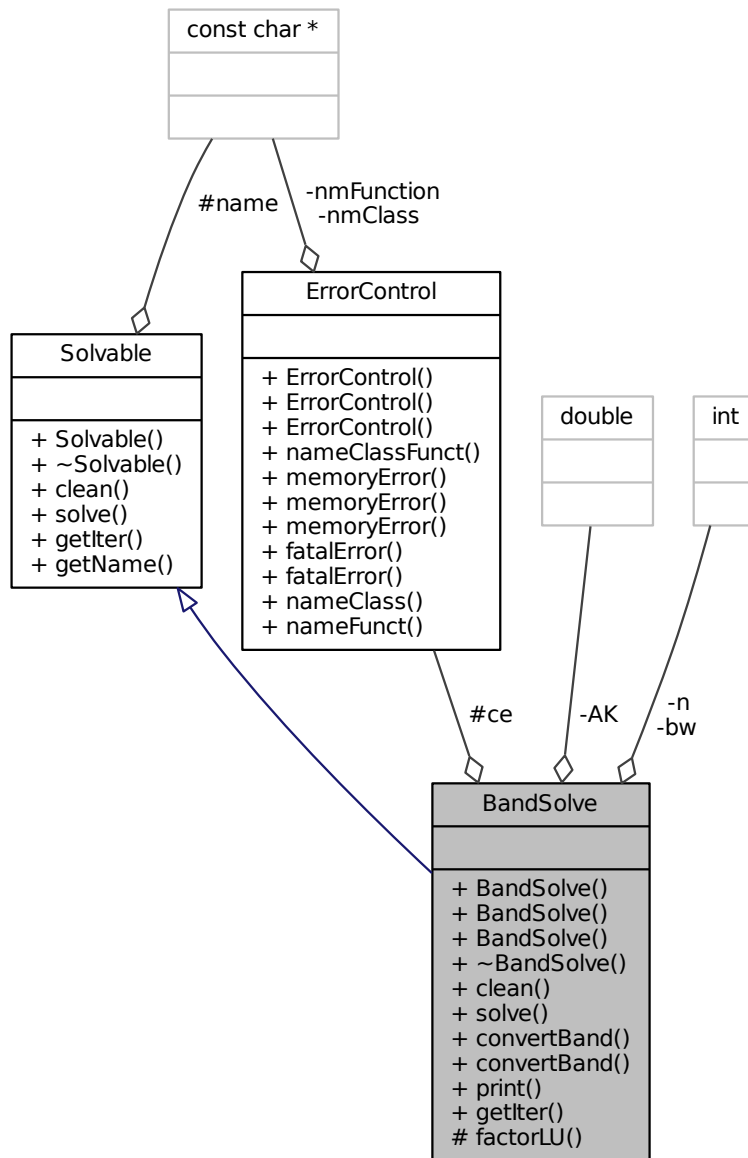
```
#include <BandSolve.hpp>
```

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]

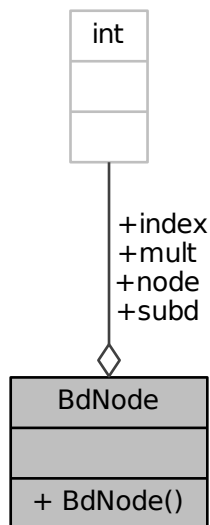
The documentation for this class was generated from the following files:

- [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`

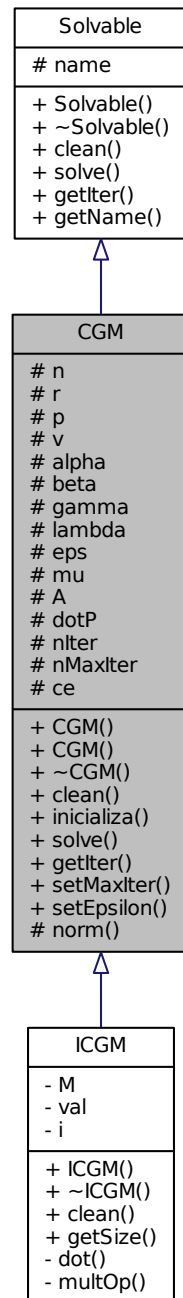
The documentation for this class was generated from the following file:

- [BdNode.hpp](#)

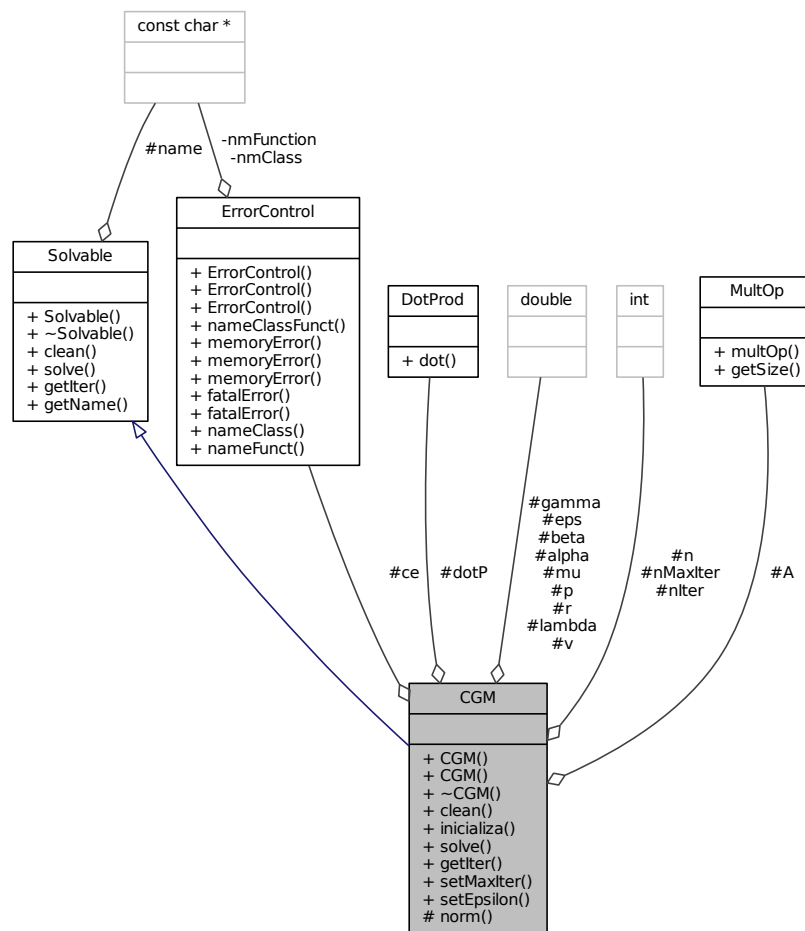
## 7.5 CGM Class Reference

```
#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 `ldouble CGM::norm( ldouble * x )` `[protected]`

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

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

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

Implements [Solvable](#).

### 7.5.3 Member Data Documentation

- 7.5.3.1 **MultOp\*** **CGM::A** [protected]
- 7.5.3.2 **Idouble** **CGM::alpha** [protected]
- 7.5.3.3 **Idouble** **CGM::beta** [protected]
- 7.5.3.4 **ErrorControl** **CGM::ce** [protected]
- 7.5.3.5 **DotProd\*** **CGM::dotP** [protected]
- 7.5.3.6 **Idouble** **CGM::eps** [protected]
- 7.5.3.7 **Idouble** **CGM::gamma** [protected]
- 7.5.3.8 **Idouble** **CGM::lambda** [protected]
- 7.5.3.9 **Idouble** **CGM::mu** [protected]
- 7.5.3.10 **int** **CGM::n** [protected]
- 7.5.3.11 **int** **CGM::nlter** [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:

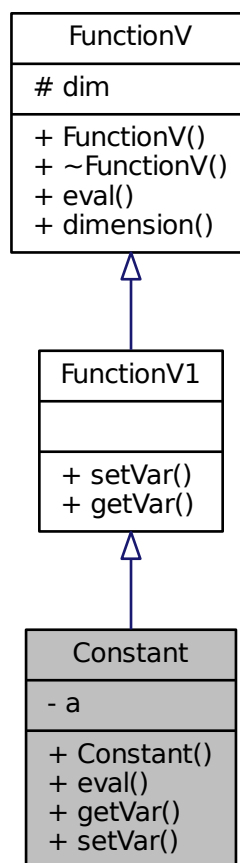
- [CGM.hpp](#)
- [CGM.cpp](#)

## 7.6 Constant Class Reference

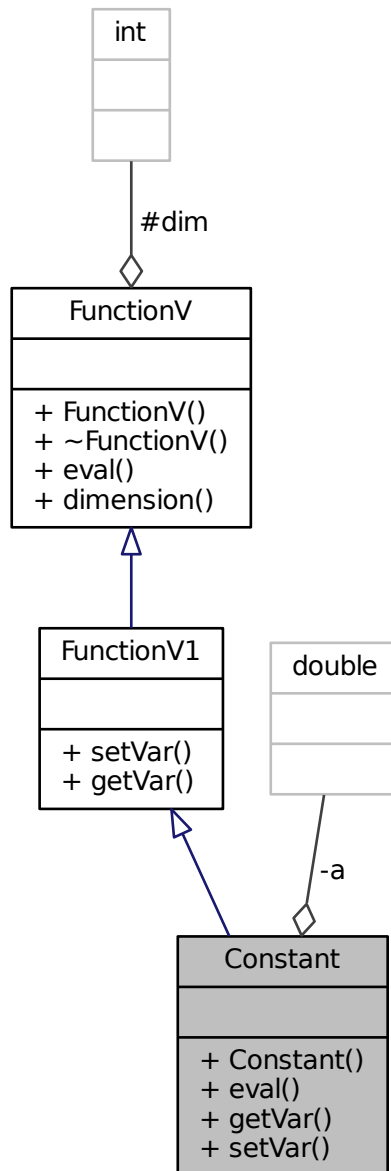
```
#include <Constant.hpp>
```



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

- [Idouble a](#)

## Additional Inherited Members

### 7.6.1 Constructor & Destructor Documentation

7.6.1.1 `Constant::Constant ( Idouble b )` `[inline]`

### 7.6.2 Member Function Documentation

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

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

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

Implements [FunctionV1](#).

### 7.6.3 Member Data Documentation

7.6.3.1 `Idouble Constant::a` `[private]`

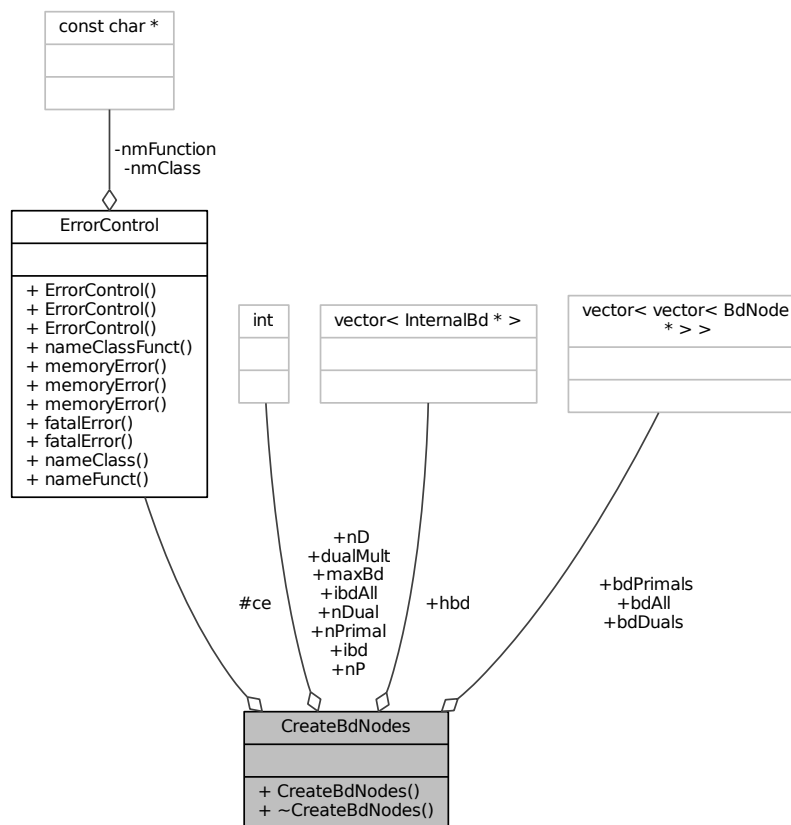
The documentation for this class was generated from the following file:

- [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`

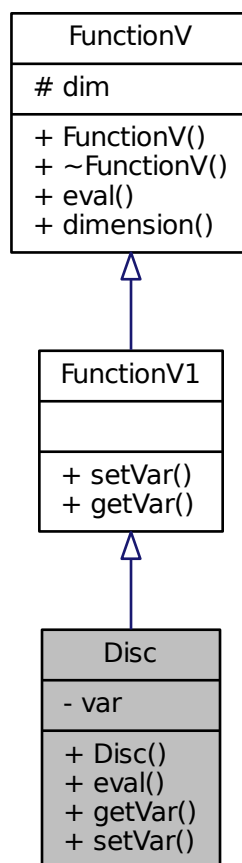
The documentation for this class was generated from the following file:

- [CreateBdNodes.hpp](#)

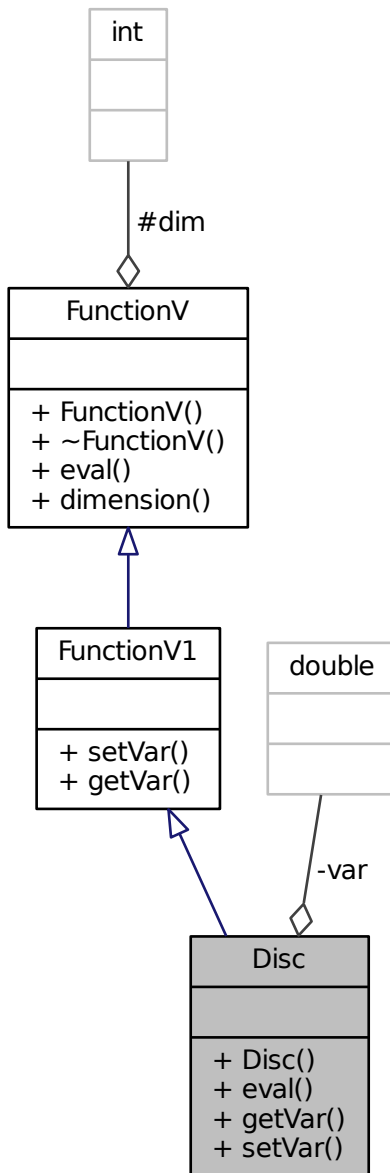
## 7.8 Disc Class Reference

```
#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:

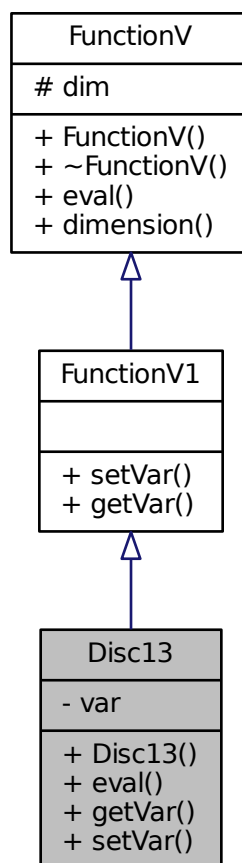
- [Disc.hpp](#)

## 7.9 Disc13 Class Reference

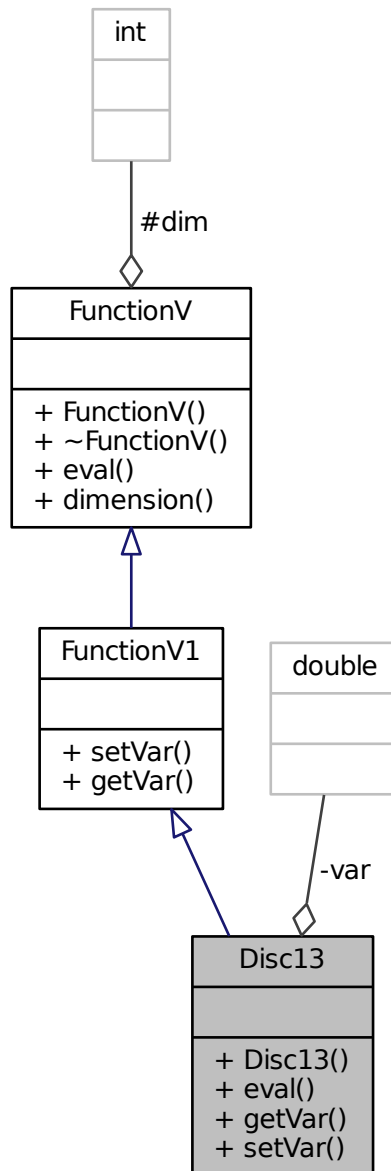
```
#include <Disc13.hpp>
```



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]`

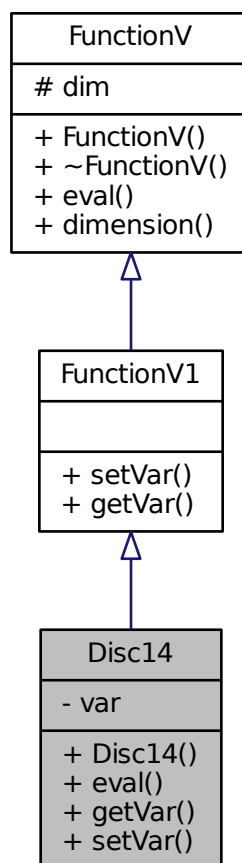
The documentation for this class was generated from the following file:

- [Disc13.hpp](#)

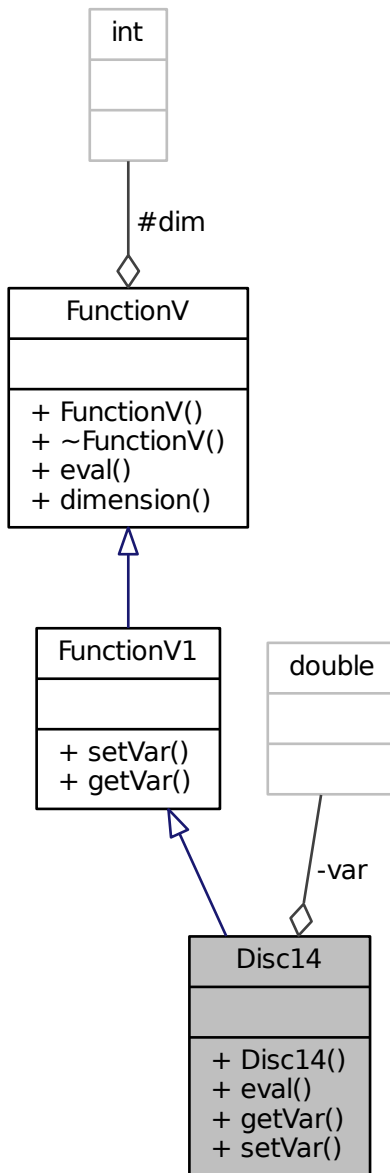
## 7.10 Disc14 Class Reference

```
#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]`

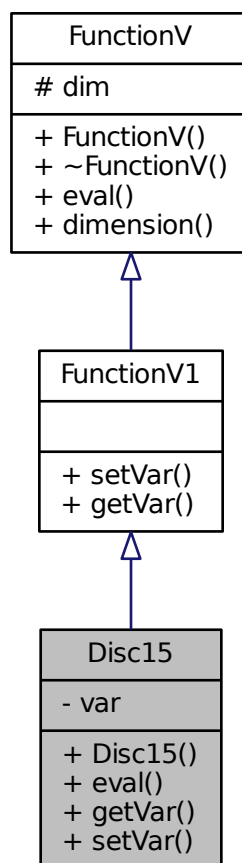
The documentation for this class was generated from the following file:

- [Disc14.hpp](#)

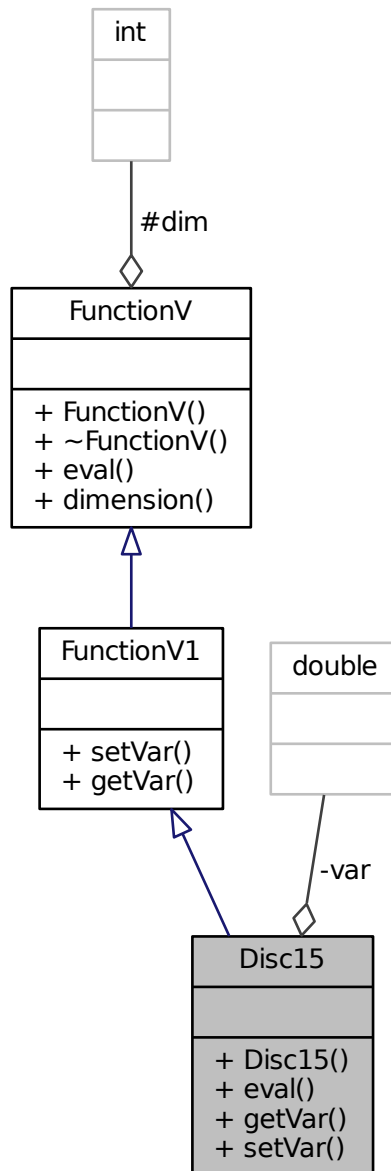
## 7.11 Disc15 Class Reference

```
#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]`

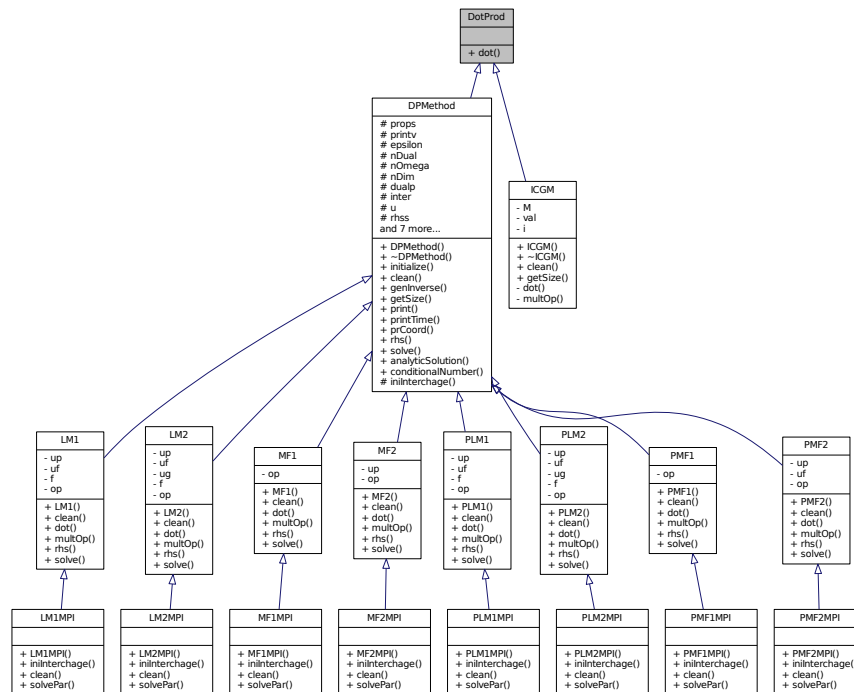
The documentation for this class was generated from the following file:

- [Disc15.hpp](#)

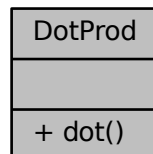
## 7.12 DotProd Class Reference

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

Inheritance diagram for DotProd:



Collaboration diagram for DotProd:



## Public Member Functions

- virtual `Idouble dot (Idouble *x, Idouble *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](#), [MF1](#), and [ICGM](#).

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

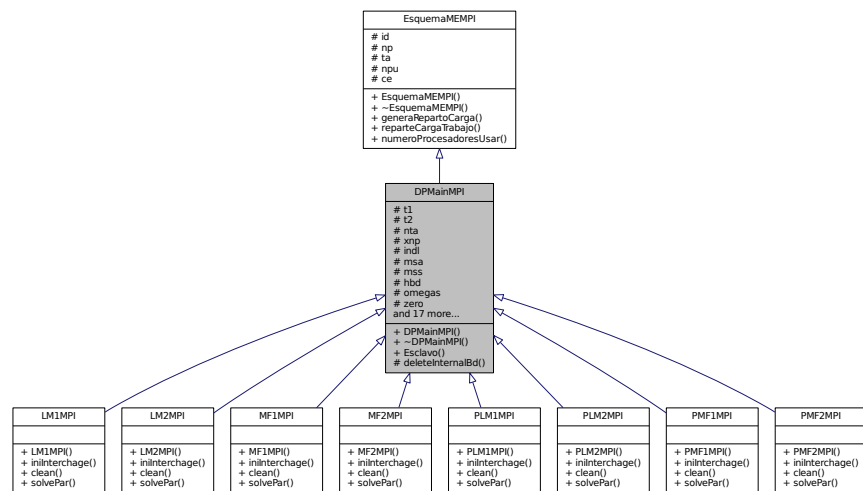
- [DotProd.hpp](#)

## 7.13 DPMainMPI Class Reference

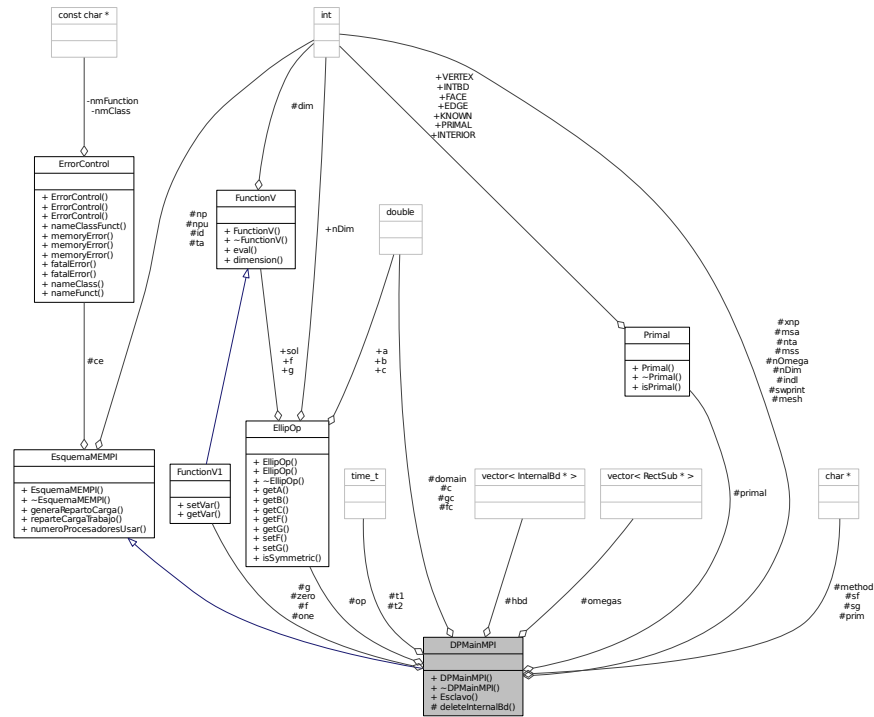
Clase base para definir a los metodos DVS-DDM.

```
#include <DPMainMPI.hpp>
```

Inheritance diagram for DPMainMPI:



Collaboration diagram for DPMainMPI:



## Public Member Functions

- **DPMainMPI** (int id, int np, PropDef &props, EllipOp &op)  
*Constructor de la clase.*
- **~DPMainMPI** ()  
*Destructor de la clase.*
- void **Esclavo** (void)  
*Esclavo.*

## Protected Member Functions

- void **deleteInternalBd** (void)

## Protected Attributes

- time\_t **t1**  
*Tiempo inicial.*
- time\_t **t2**  
*Tiempo final.*
- int **nta**  
*Nmero de tareas por nodo esclavo.*

- int [xnp](#)  
*Nmero de esclavo en el que estara la tarea.*
- int [indl](#)  
*Nmero de tarea dentro del esclavo.*
- int [msa](#) [10]  
*Arreglo para recibir mensajes.*
- int [mss](#) [10]  
*Arreglo para enviar mensajes.*
- vector< [InternalBd](#) \* > [hbd](#)
- vector< [RectSub](#) \* > [omegas](#)
- [FunctionV1](#) \* [zero](#)
- [FunctionV1](#) \* [one](#)
- [FunctionV1](#) \* [f](#)
- [FunctionV1](#) \* [g](#)
- char \* [sf](#)
- char \* [sg](#)
- [ldouble](#) [fc](#)
- [ldouble](#) [gc](#)
- int \* [mesh](#)
- char \* [prim](#)
- char \* [method](#)
- int [swprint](#)
- [ldouble](#) \*\* [domain](#)
- [ldouble](#) [c](#)
- [Primal](#) \* [primal](#)
- [EllipOp](#) \* [op](#)
- int [nDim](#)
- int [nOmega](#)

### 7.13.1 Detailed Description

Clase base para definir a los metodos DVS-DDM.

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

#### Author

Antonio Carrillo Ledesma

#### Date

primavera 2010

#### Version

1.0.0

**Bug** No hay errores conocidos

### 7.13.2 Constructor & Destructor Documentation

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

Constructor de la clase.

#### 7.13.2.2 DPMainMPI::~~DPMainMPI ( )

Destructor de la clase.

### 7.13.3 Member Function Documentation

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

#### 7.13.3.2 void DPMainMPI::Esclavo ( void )

Esclavo.

### 7.13.4 Member Data Documentation

#### 7.13.4.1 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]

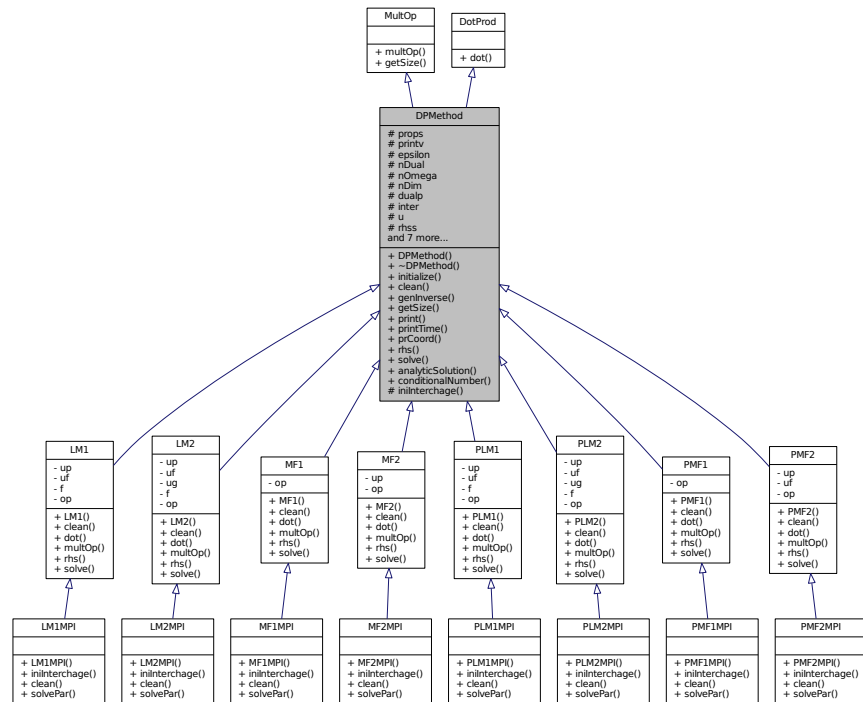
The documentation for this class was generated from the following files:

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

## 7.14 DPMMethod Class Reference

```
#include <DPMMethod.hpp>
```

Inheritance diagram for DPMMethod:





[illegible]

- **DPMMethod** (**PropDef** &**props**)
- virtual **~DPMMethod** ()
- void **initialize** (void)
- virtual void **clean** (void)=0
- void **genInverse** (int type)
- int **getSize** (void)  
*vector size*
- void **print** (**ldouble** \*u)
- void **printTime** (void)
- const char \* **prCoord** (**ldouble** \*x)
- virtual void **rhs** (void)=0
- virtual void **solve** (void)=0
- double **analyticSolution** (double \*x)

- void [conditionalNumber](#) (bool symetric)  
*Calcula el numero de condicionamiento.*

## Protected Member Functions

- virtual void [iniInterchage](#) (void)  
*Inicializa los subdominios.*

## Protected Attributes

- [PropDef](#) \* [props](#)
- int [printv](#)
- [ldouble](#) [epsilon](#)
- int [nDual](#)
- int [nOmega](#)
- int [nDim](#)
- [DualPrimal](#) \* [dualp](#)
- [Interchange](#) \* [inter](#)
- [ldouble](#) \* [u](#)
- [ldouble](#) \* [rhss](#)
- [ldouble](#) \* [scr](#)
- [Solvable](#) \* [solver](#)
- time\_t [time0](#)
- time\_t [time1](#)
- time\_t [time2](#)
- time\_t [time3](#)
- [ErrorControl](#) [ce](#)

*Control de errores.*

## 7.14.1 Constructor & Destructor Documentation

7.14.1.1 `DPMethoD::DPMethoD ( PropDef & props )` [inline]

7.14.1.2 `virtual DPMethoD::~~DPMethoD ( )` [inline],[virtual]

## 7.14.2 Member Function Documentation

7.14.2.1 `double DPMethoD::analyticSolution ( double * x )`

7.14.2.2 `virtual void DPMethoD::clean ( void )` [pure virtual]

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

7.14.2.3 `void DPMethoD::conditionalNumber ( bool symetric )`

Calcula el numero de condicionamiento.

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

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

vector size

Implements [MultOp](#).

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

Inicializa los subdominios.

Reimplemented in [LM2MPI](#), [MF1MPI](#), [MF2MPI](#), [PLM1MPI](#), [PLM2MPI](#), [PMF2MPI](#), [LM1MPI](#), and [PMF1MPI](#).

7.14.2.7 void DPMethod::initialize ( void )

7.14.2.8 const char \* DPMethod::prCoord ( 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\*** DPMethod::dualp [protected]

7.14.3.3 **Idouble** DPMethod::epsilon [protected]

7.14.3.4 **Interchange\*** DPMethod::inter [protected]

7.14.3.5 **int** DPMethod::nDim [protected]

7.14.3.6 **int** DPMethod::nDual [protected]

7.14.3.7 **int** DPMethod::nOmega [protected]

7.14.3.8 **int** DPMethod::printv [protected]

7.14.3.9 **PropDef\*** `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]

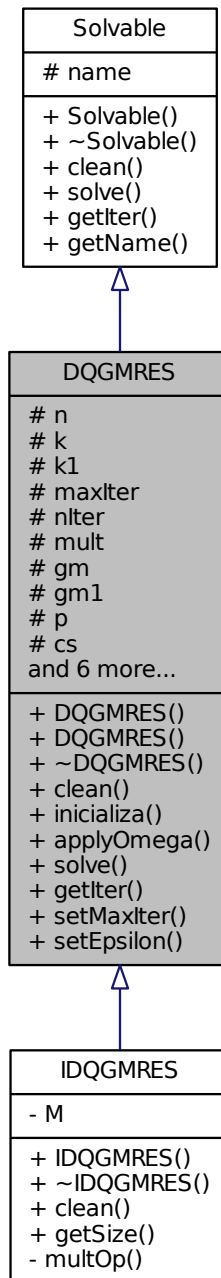
The documentation for this class was generated from the following files:

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

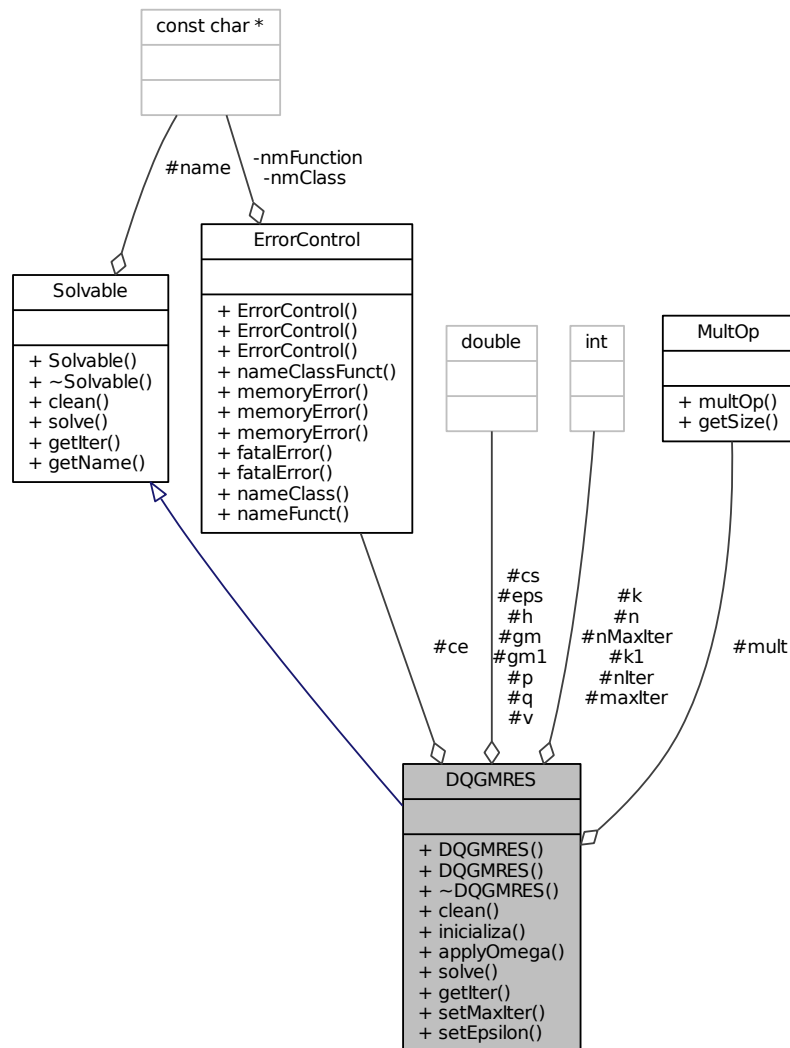
## 7.15 DQGMRES Class Reference

```
#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 ( ldouble ep )` `[inline]`

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

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

Implements [Solvable](#).

### 7.15.3 Member Data Documentation

7.15.3.1 **ErrorControl** DQGMRES::ce [protected]

7.15.3.2 **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>
```





- Generated on Tue Sep 29 2015 08:06:05 for DDM-DVS by Doxygen

## Protected Attributes

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

*Control de errores.*

## 7.16.1 Constructor & Destructor Documentation

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

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

## 7.16.2 Member Function Documentation

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

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

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

7.16.2.4 `void DualPrimal::genMats ( void )`

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

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

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

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

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

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

## 7.16.3 Member Data Documentation

7.16.3.1 `ErrorControl DualPrimal::ce` `[protected]`

Control de errores.

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

7.16.3.3 `int DualPrimal::nD` `[protected]`

7.16.3.4 `int DualPrimal::nDual` [protected]

7.16.3.5 `int DualPrimal::nP` [protected]

7.16.3.6 `int DualPrimal::nPrimal` [protected]

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

7.16.3.8 `Idouble* DualPrimal::XP` [protected]

7.16.3.9 `Idouble* DualPrimal::YP` [protected]

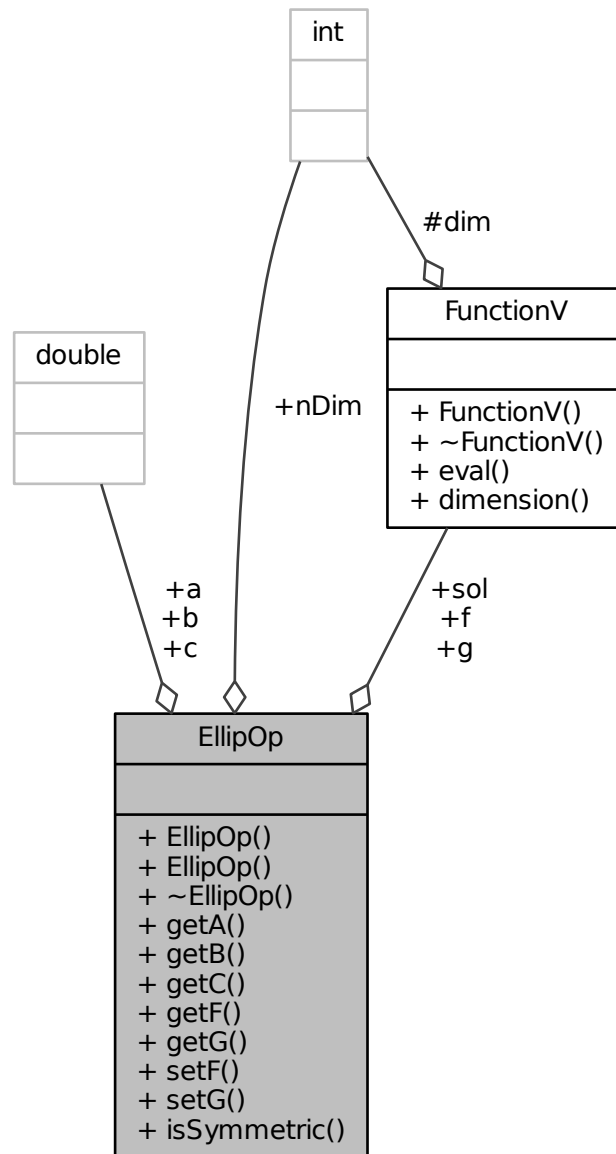
The documentation for this class was generated from the following files:

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

## 7.17 EllipOp Class Reference

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

Collaboration diagram for EllipOp:



### Public Member Functions

- `EllipOp` (`int nDim`, `ldouble *a`, `ldouble *b`, `ldouble c`, `FunctionV &f`, `FunctionV &g`, `FunctionV &sol`)
- `EllipOp` (`int nDim`, `ldouble *a`, `ldouble *b`, `ldouble c`)
- `~EllipOp` ()
- `ldouble * getA` (void)

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

### Public Attributes

- `int nDim`
- `ldouble * a`
- `ldouble * b`
- `ldouble c`
- `FunctionV * f`
- `FunctionV * g`
- `FunctionV * sol`

### 7.17.1 Constructor & Destructor Documentation

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

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

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

### 7.17.2 Member Function Documentation

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

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

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

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

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

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

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

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

### 7.17.3 Member Data Documentation

7.17.3.1 `ldouble* EllipOp::a`

7.17.3.2 `ldouble* EllipOp::b`

7.17.3.3 `Idouble EllipOp::c`

7.17.3.4 `FunctionV* EllipOp::f`

7.17.3.5 `FunctionV* EllipOp::g`

7.17.3.6 `int EllipOp::nDim`

7.17.3.7 `FunctionV* EllipOp::sol`

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

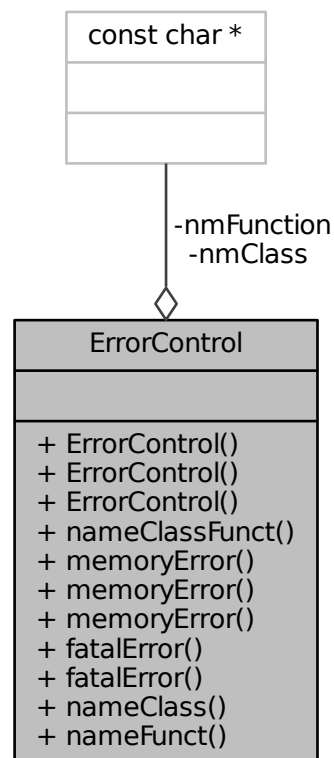
- [EllipOp.hpp](#)

## 7.18 ErrorControl Class Reference

Error Control.

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

Collaboration diagram for ErrorControl:



## Public Member Functions

- [ErrorControl](#) (void)
- [ErrorControl](#) (const char \*clas)
- [ErrorControl](#) (const char \*clas, const char \*fun)
- void [nameClassFunct](#) (const char \*clas, const char \*func)
- void [memoryError](#) (const char \*var)
- void [memoryError](#) (const char \*var, int i)
- void [memoryError](#) (const char \*var, const char \*func)
- void [fatalError](#) (int cod)
- void [fatalError](#) (int cod, const char \*txt)
- void [nameClass](#) (const char \*clas)
- void [nameFunct](#) (const char \*func)

## Private Attributes

- const char \* [nmClass](#)  
*Name of class.*
- const char \* [nmFunction](#)  
*Name of function.*

### 7.18.1 Detailed Description

Error Control.

#### Author

Antonio Carrillo

#### Date

Winter 2010

#### Version

0.0.1

**Bug** No errors detected

**Todo** Exception handling

### 7.18.2 Constructor & Destructor Documentation

#### 7.18.2.1 ErrorControl::ErrorControl ( void )

Class Constructor

#### 7.18.2.2 ErrorControl::ErrorControl ( const char \* clas )

Class Constructor

## Parameters

<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::nameClassFunct ( 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::nameFunct ( 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:

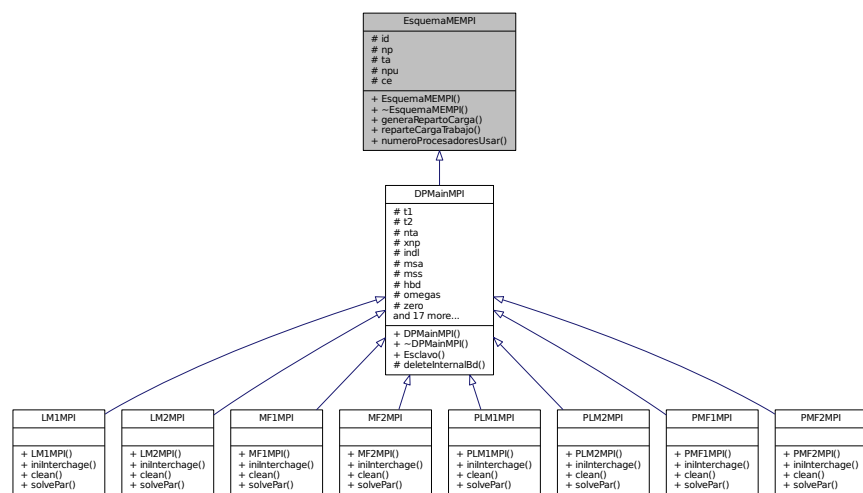
- [ErrorControl.hpp](#)
- [ErrorControl.cpp](#)

## 7.19 EsquemaMEMPI Class Reference

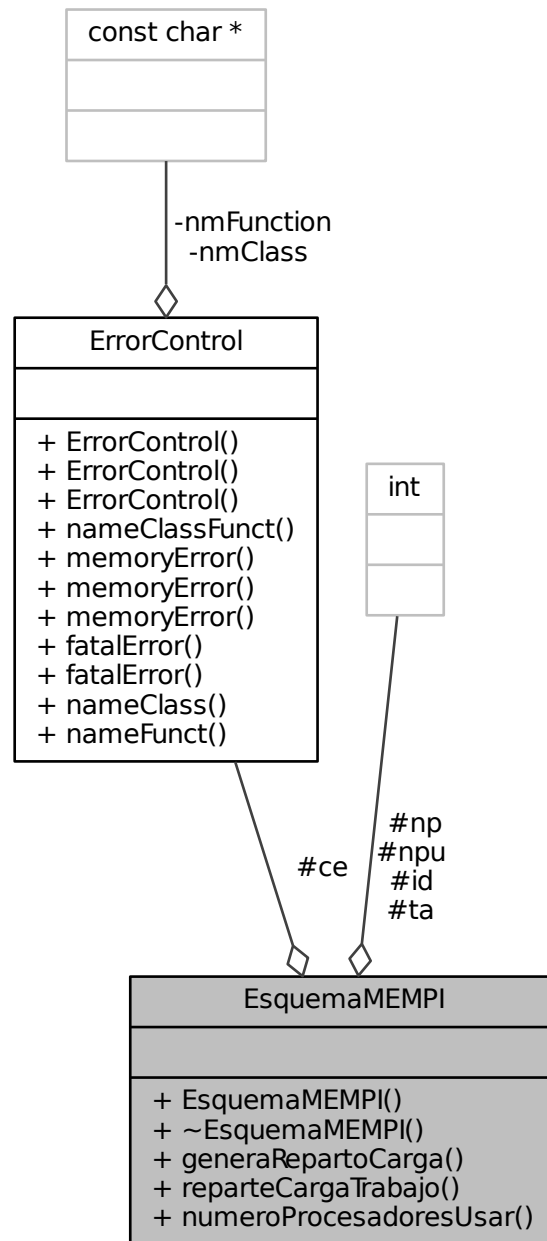
Clase base para definir el Esquema Maestro-Eslavo en MPI.

```
#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.

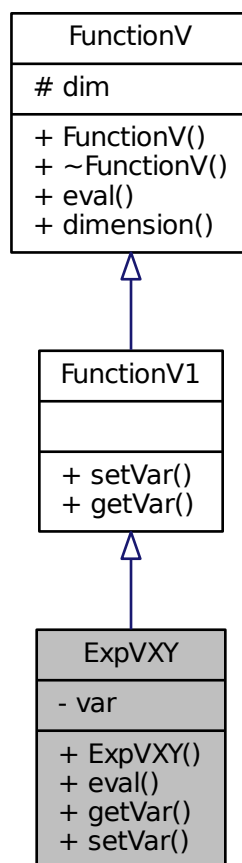
The documentation for this class was generated from the following files:

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

## 7.20 ExpVXY Class Reference

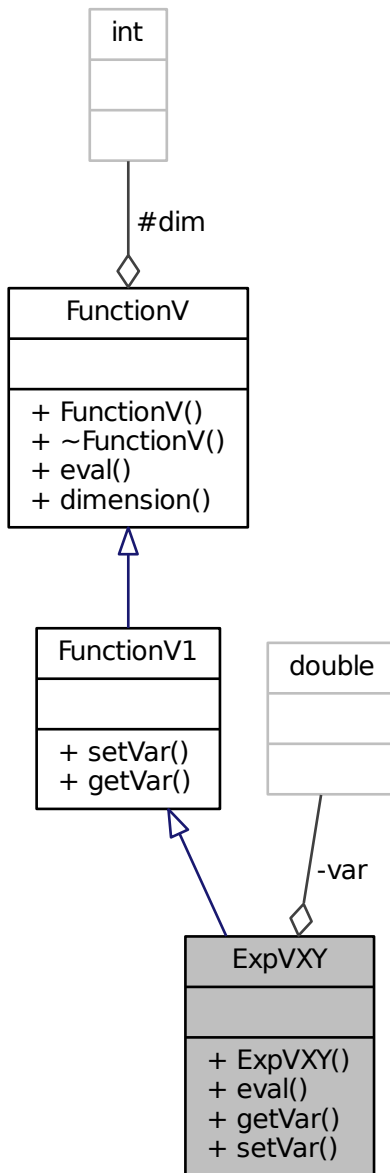
```
#include <ExpVXY.hpp>
```

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]`

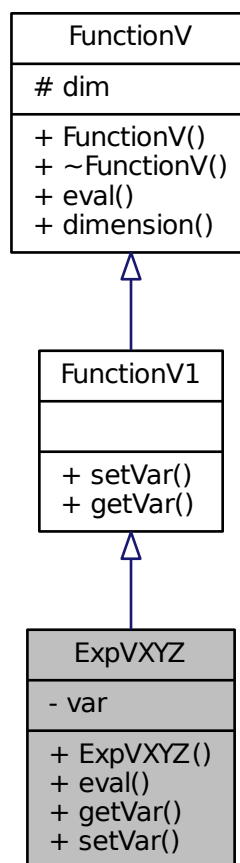
The documentation for this class was generated from the following file:

- [ExpVXY.hpp](#)

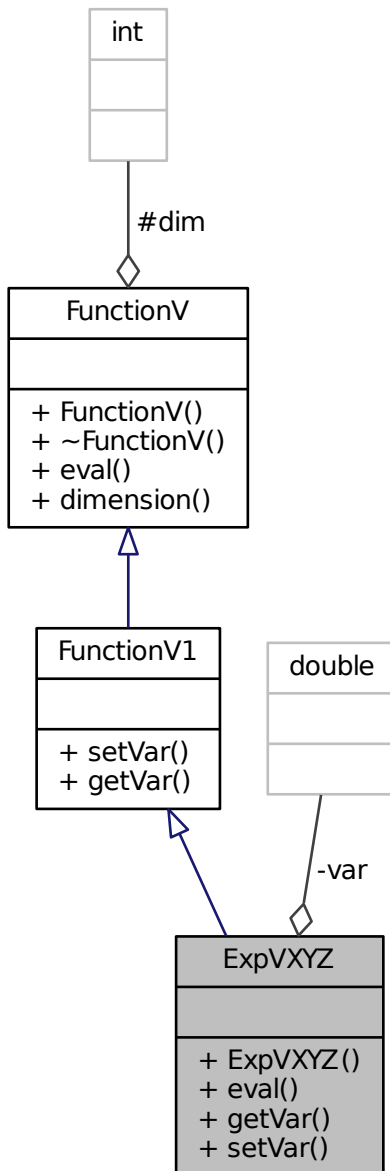
## 7.21 ExpVXYZ Class Reference

```
#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]`

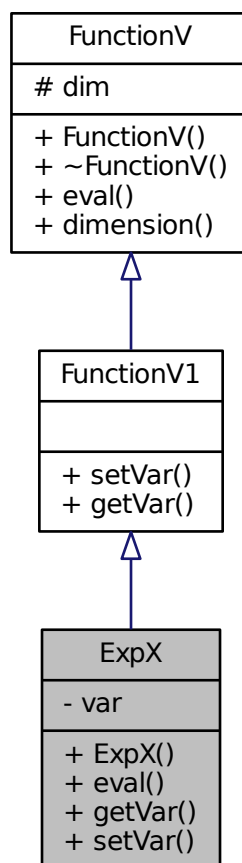
The documentation for this class was generated from the following file:

- [ExpVXYZ.hpp](#)

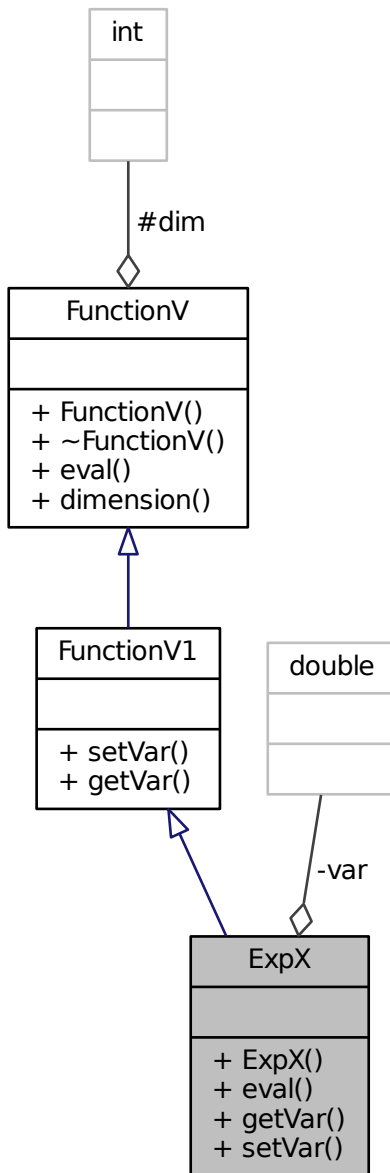
## 7.22 ExpX Class Reference

```
#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:

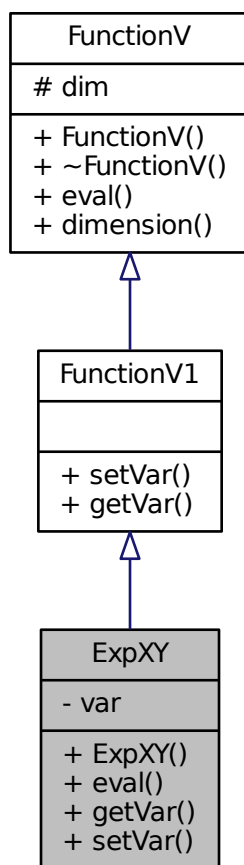
- [ExpX.hpp](#)

## 7.23 ExpXY Class Reference

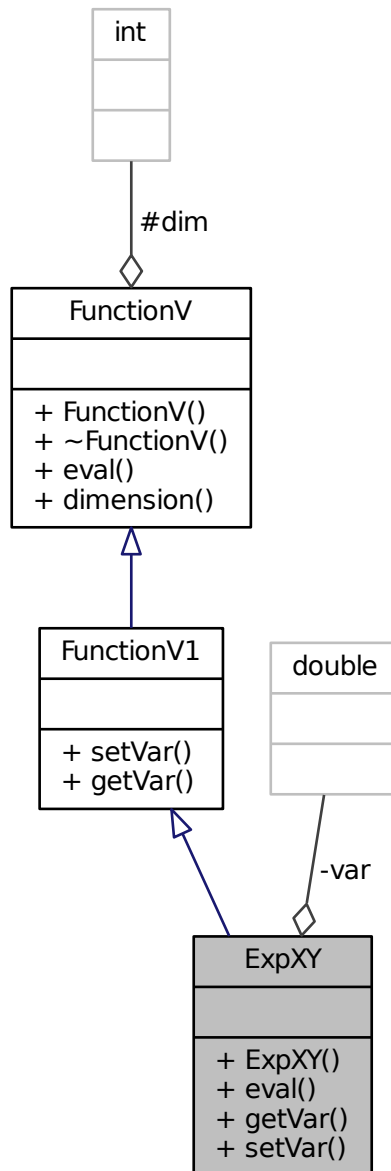
```
#include <ExpXY.hpp>
```



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

- [Idouble var](#)

## Additional Inherited Members

### 7.23.1 Constructor & Destructor Documentation

7.23.1.1 `ExpXY::ExpXY ( Idouble b )` `[inline]`

### 7.23.2 Member Function Documentation

7.23.2.1 `Idouble ExpXY::eval ( int d, Idouble * x )` `[inline]`, `[virtual]`

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

7.23.2.3 `void ExpXY::setVar ( Idouble b )` `[inline]`, `[virtual]`

Implements [FunctionV1](#).

### 7.23.3 Member Data Documentation

7.23.3.1 `Idouble ExpXY::var` `[private]`

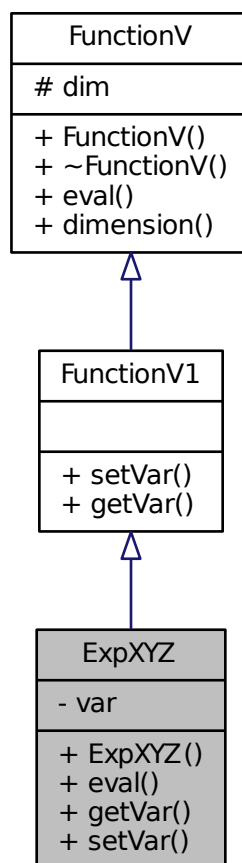
The documentation for this class was generated from the following file:

- [ExpXY.hpp](#)

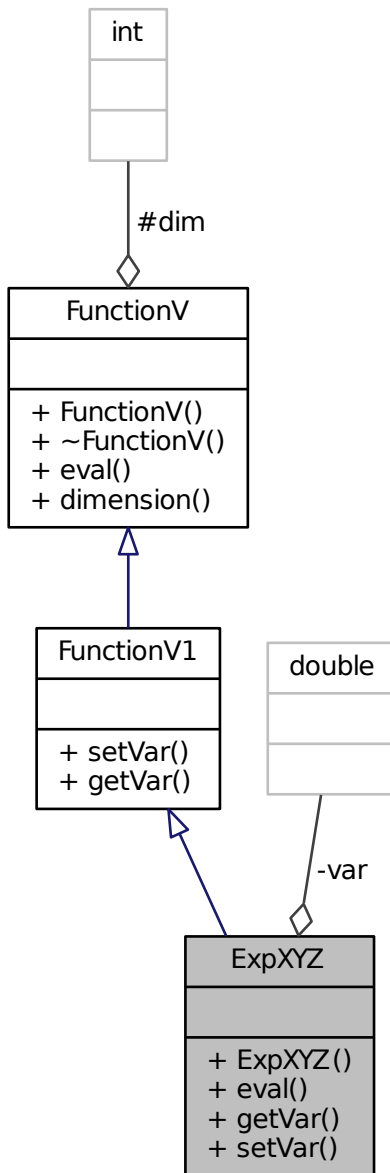
## 7.24 ExpXYZ Class Reference

```
#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]`

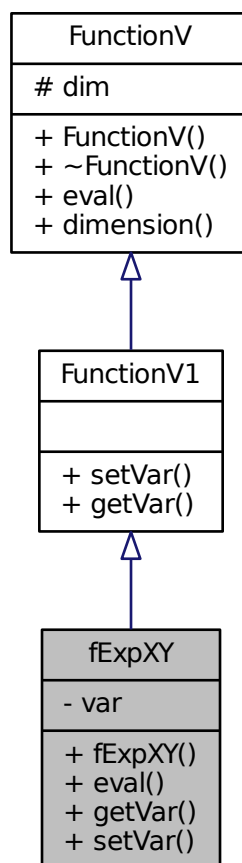
The documentation for this class was generated from the following file:

- [ExpXYZ.hpp](#)

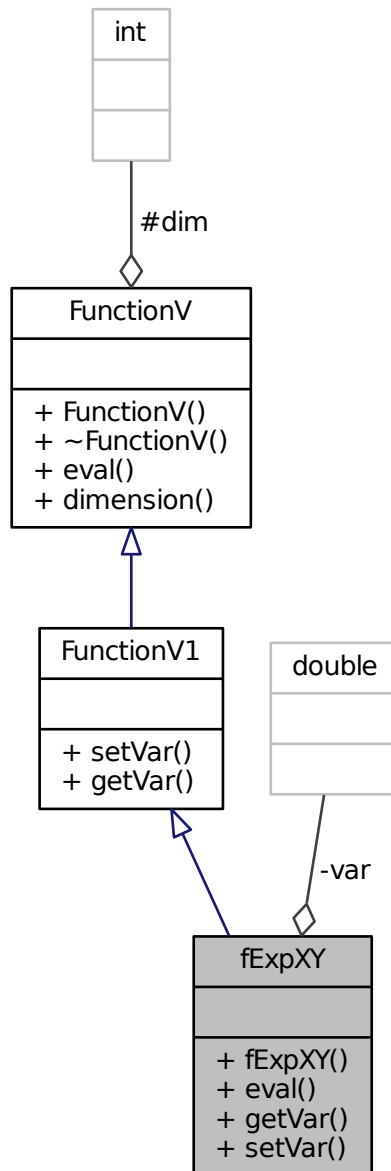
## 7.25 fExpXY Class Reference

```
#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]`

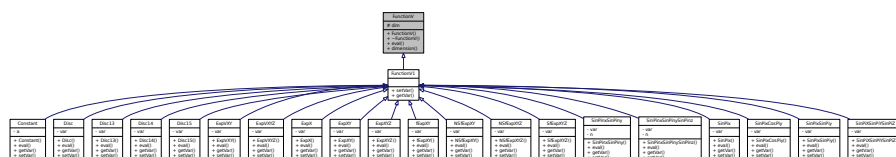
The documentation for this class was generated from the following file:

- [fExpXY.hpp](#)

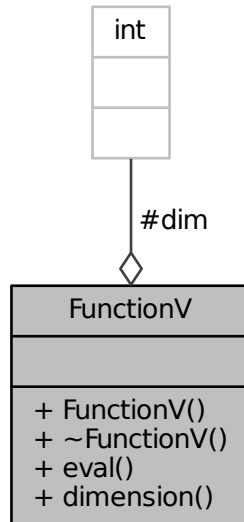
## 7.26 FunctionV Class Reference

```
#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]

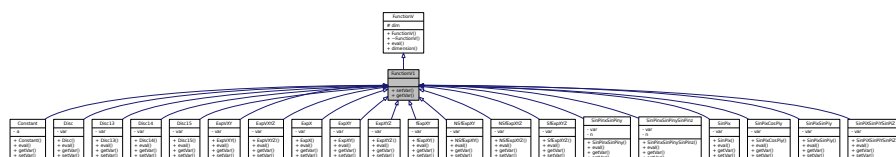
The documentation for this class was generated from the following file:

- [FunctionV.hpp](#)

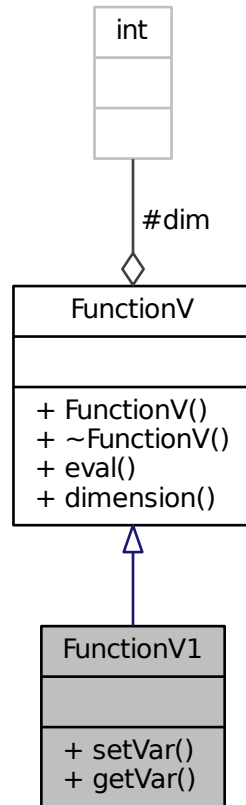
## 7.27 FunctionV1 Class Reference

```
#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 ( ldouble 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](#).

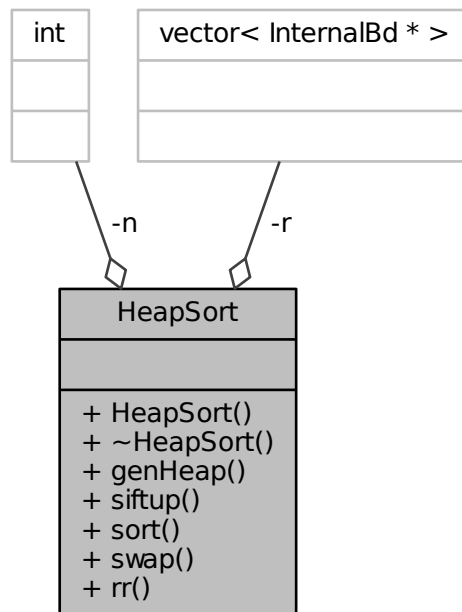
The documentation for this class was generated from the following file:

- [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:

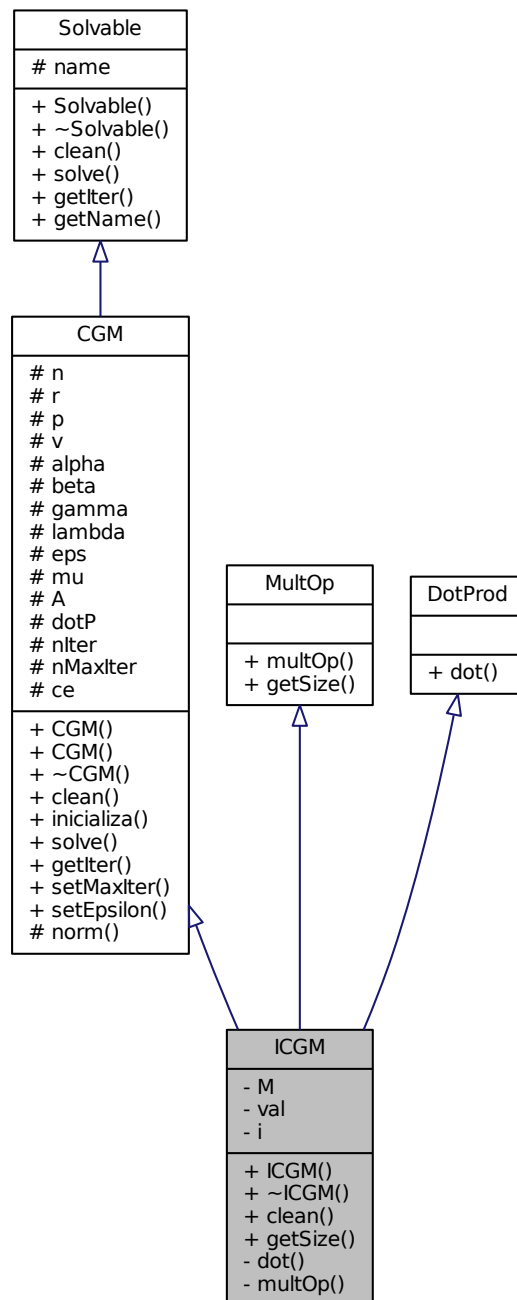
- [HeapSort.hpp](#)

## 7.29 ICGM Class Reference

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

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

Inheritance diagram for ICGM:







*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`)

*Multiplifica  $Au=v$ .*

### Private Attributes

- `MatrizDispersa * M`

*Multiplifica  $Au=v$ .*

- `ldouble val`

*Variables temporales.*

- int `i`

### Additional Inherited Members

#### 7.29.1 Detailed Description

Clase para implementar `CGM` con matrices bandadas o dispersas.

##### Author

Antonio Carrillo Ledesma

##### Date

primavera 2010

##### Version

1.0.1

**Bug** No hay errores conocidos

#### 7.29.2 Constructor & Destructor Documentation

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

Contructor de la clase.

7.29.2.2 `ICGM::~~ICGM ( )` `[inline]`

Destructor de la clase.

### 7.29.3 Member Function Documentation

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

Reimplemented from [CGM](#).

7.29.3.2 `Idouble ICGM::dot ( Idouble * u, Idouble * v )` `[inline],[private],[virtual]`

Producto punto.

Implements [DotProd](#).

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

vector size

Implements [MultOp](#).

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

Multiplica Au=v.

Implements [MultOp](#).

### 7.29.4 Member Data Documentation

7.29.4.1 `int ICGM::i` `[private]`

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

Multiplica Au=v.

7.29.4.3 `Idouble ICGM::val` `[private]`

Variables temporales.

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

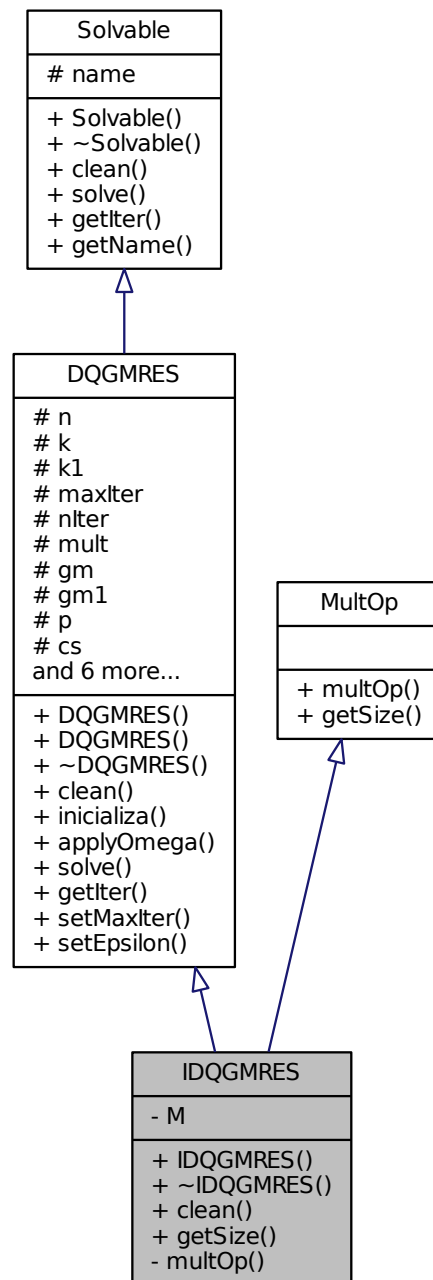
- [ICGM.hpp](#)

## 7.30 IDQGMRES Class Reference

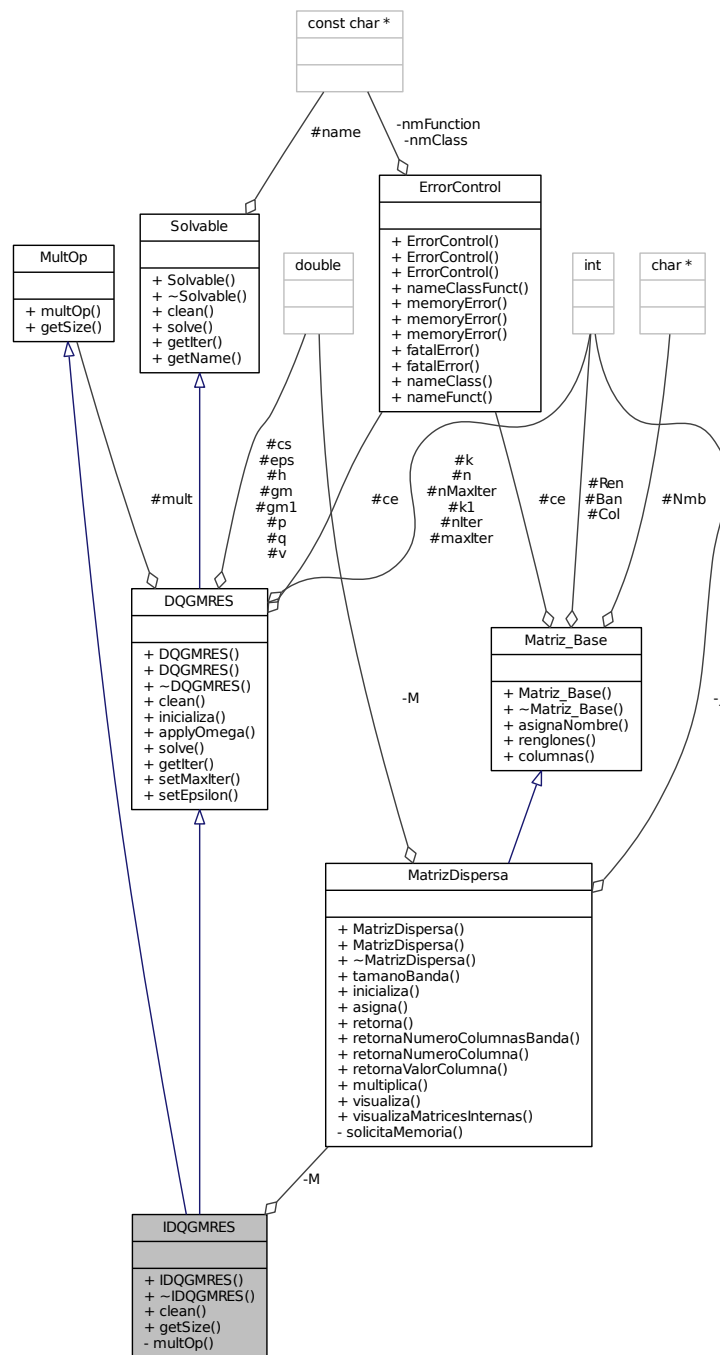
Clase para implementar [DQGMRES](#) con matrices bandadas o dispersas.

```
#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](#) (Idouble \*u, Idouble \*v)

*Multiplca  $Au=v$ .*

### Private Attributes

- [MatrizDispersa](#) \* [M](#)

*Matriz Bandada o Dispersa.*

### Additional Inherited Members

#### 7.30.1 Detailed Description

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

#### Author

Antonio Carrillo Ledesma

#### Date

primavera 2010

#### Version

1.0.1

**Bug** No hay errores conocidos

#### 7.30.2 Constructor & Destructor Documentation

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

Constructor de la clase.

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

#### 7.30.3 Member Function Documentation

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

Reimplemented from [DQGMRES](#).

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

vector size

Implements [MultOp](#).

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

Multiplica  $Au=v$ .

Implements [MultOp](#).

## 7.30.4 Member Data Documentation

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

Matriz Bandada o Dispersa.

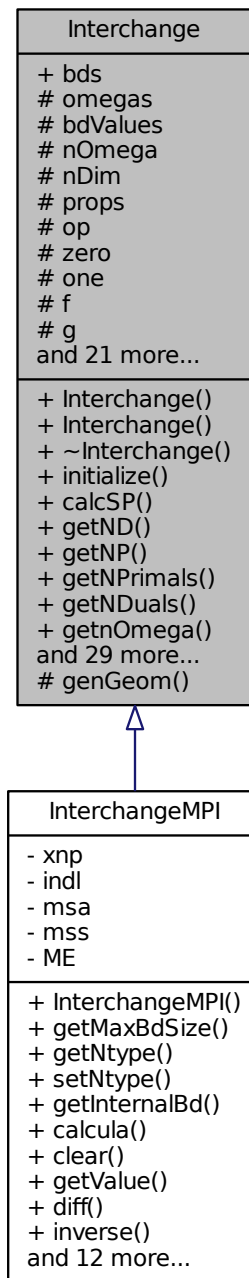
The documentation for this class was generated from the following file:

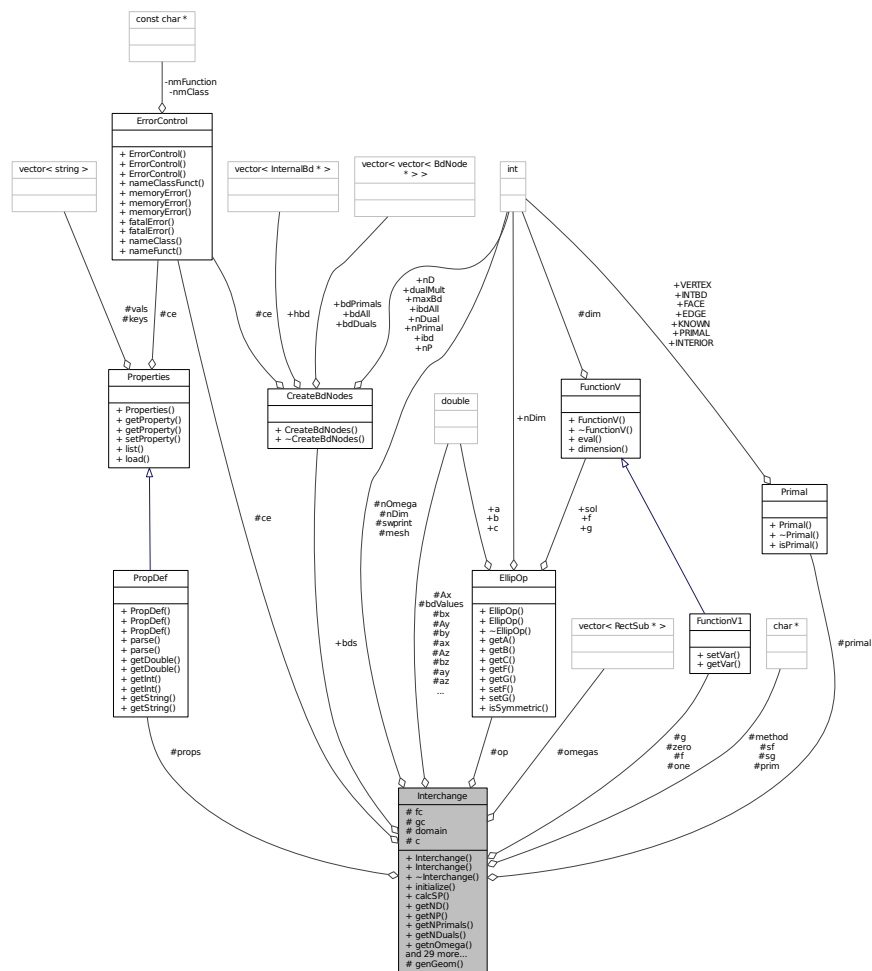
- [IDQGMRES.hpp](#)

## 7.31 Interchange Class Reference

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

Inheritance diagram for Interchange:





## 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 **getNPPrimals** (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)
 

$scr[sc][ ] = \text{Dirichlet boundary values of all subdomains}$
- virtual void `multOp` (int sc1, int sc2)
 

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

$scr[sc][ ] = \text{initial right-hand-side (all subdomains)}$
- 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 Idouble Interchange::getValue ( int e, int scr, int node ) [inline],[virtual]`

7.31.2.21 `virtual Idouble 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 Idouble 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, Idouble 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]`

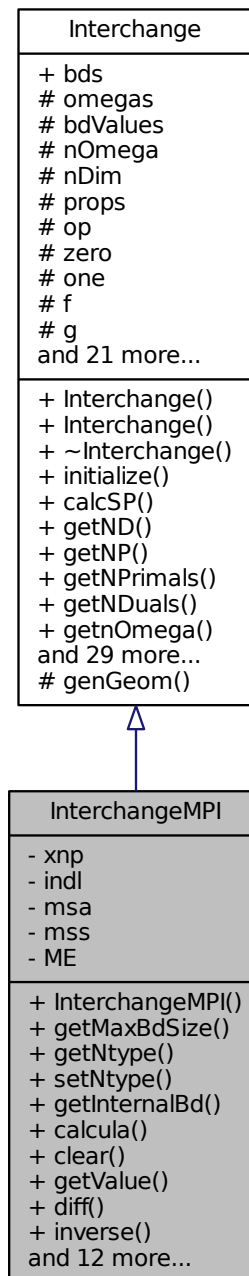
The documentation for this class was generated from the following files:

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

## 7.32 InterchangeMPI Class Reference

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

Inheritance diagram for InterchangeMPI:





The diagram illustrates the class structure and relationships for the 'interchange' package. Key components include:

- Classes and their methods:**
  - ErrorControl:** + ErrorControl(), + ErrorControl(), + nameClassFunc(), + memoryError(), + memoryError(), + fatalError(), + fatalError(), + nameClass(), + nameFunc().
  - FunctionV:** + FunctionV(), + ~FunctionV(), + eval(), + dimension().
  - FunctionV1:** + setVar(), + getVar().
  - Primal:** + Primal(), + ~Primal(), + isPrimal().
  - Interchange:** + interchange(), + interchange(), + ~interchange(), + initialize(), + calcSP(), + getND(), + getNP(), + getPrimals(), + getNDuals(), + getNOmega(), + getOmega(), + 29 more..., + genGeom().
  - InterchangeMPI:** + InterchangeMPI(), + getMaxBdSize(), + getNtype(), + setNtype(), + getInternalBd(), + calcula(), + clear(), + getValue(), + diff(), + inverse(), + 12 more...
  - EsquemaMEMPI:** + EsquemaMEMPI(), + ~EsquemaMEMPI(), + generaRepartoCarga(), + reparteCargaTrabajo(), + numeraProcesadoresUsar().
  - CreateBdNodes:** + CreateBdNodes(), + ~CreateBdNodes().
  - EllipOp:** + EllipOp(), + EllipOp(), + ~EllipOp(), + getA(), + getB(), + getC(), + getF(), + getG(), + setF(), + setG(), + isSymmetric().
  - PropDef:** + PropDef(), + PropDef(), + parse(), + parse(), + getDouble(), + getDouble(), + getInt(), + getInt(), + getString(), + getString().
  - Properties:** + Properties(), + getProperty(), + getProperty(), + setProperty(), + list(), + load().
- Relationships:**
  - Inheritance:** Indicated by solid arrows with open heads. Examples: InterchangeMPI inherits from Interchange; EllipOp inherits from FunctionV.
  - Associations:** Indicated by dashed arrows. Examples: Interchange is associated with FunctionV, Primal, and EllipOp.
  - Generalizations:** Indicated by solid arrows. Examples: Interchange is a generalization of InterchangeMPI.
- Attributes and Constants:** Various attributes like #dim, #bdValues, #g, #zero, #f, #one, #op, #ce, #vals, #keys, #props, #method, #sf, #sg, #prim, #omega, #method, #sf, #sg, #prim, #bds, #fc, #gc, #domain, #c are shown across different classes.

- **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, Idouble * 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.

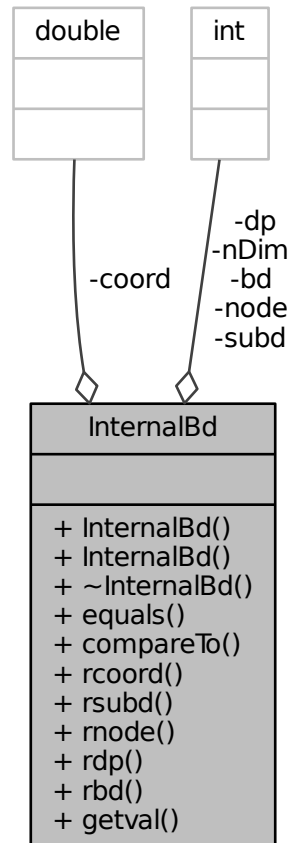
The documentation for this class was generated from the following files:

- [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` [\[private\]](#)

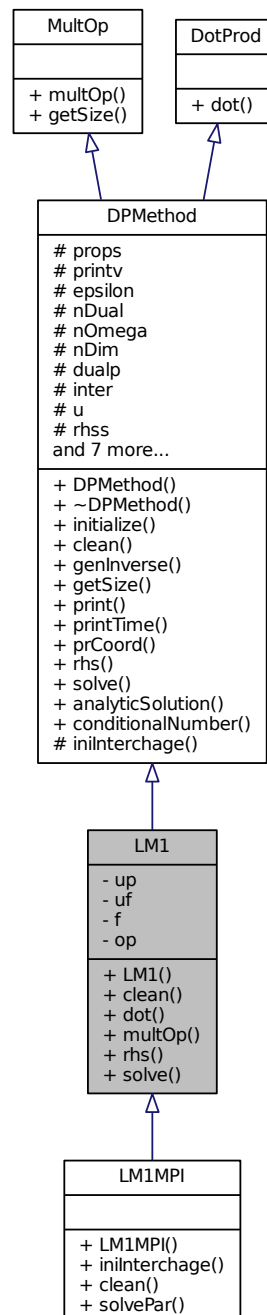
The documentation for this class was generated from the following file:

- [InternalBd.hpp](#)

## 7.34 LM1 Class Reference

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

Inheritance diagram for LM1:





[illegible]

- **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]`

$y = A \cdot x$

Implements [MultOp](#).

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

Implements [DPMethod](#).

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

Implements [DPMethod](#).

### 7.34.3 Member Data Documentation

7.34.3.1 `ldouble* LM1::f` `[private]`

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

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

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

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

- [LM1.hpp](#)

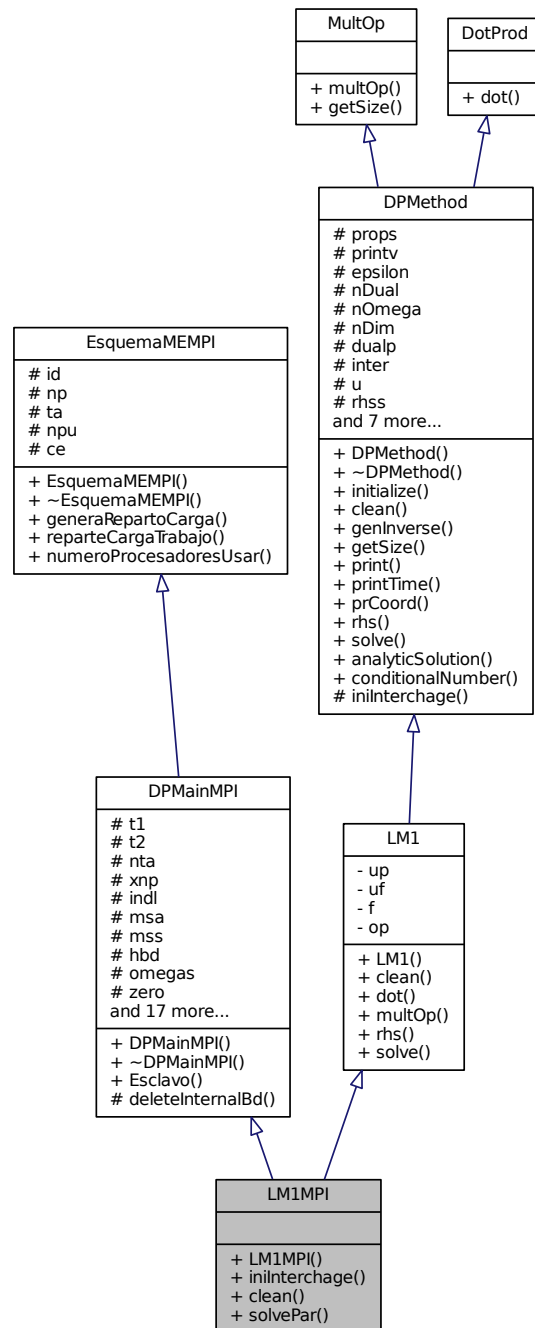
- [LM1.cpp](#)

## 7.35 LM1MPI Class Reference

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

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

Inheritance diagram for LM1MPI:



- **LM1MPI** (int **id**, int **np**, **PropDef** &**props**, **EllipOp** &**op**)  
*Constructor de la clase.*
- void **iniInterchange** (void)  
*Inicializa **InterchangeMPI** en lugar de **Interchange**.*
- void **clean** (void)
- void **solvePar** (void)  
*Sobrecarga de la aplicacion.*

Generated on Tue Sep 29 2015 08:06:05 for DDM-DVS by Doxygen

### 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 [DPMMethod](#).

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

Sobrecarga del la aplicacion.

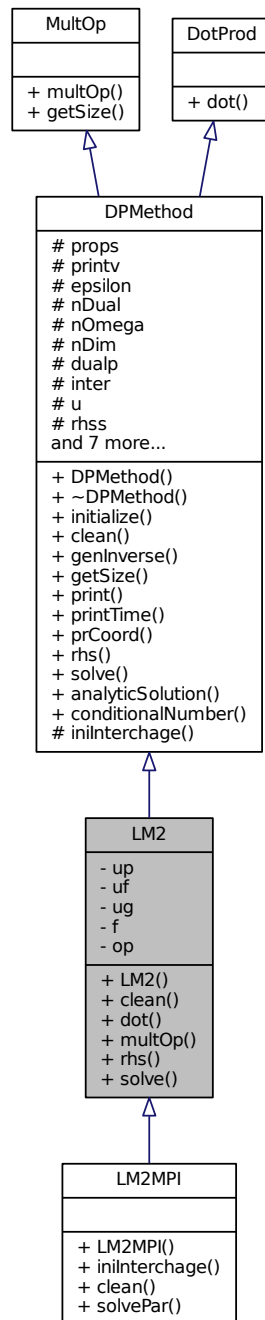
The documentation for this class was generated from the following file:

- [LM1MPI.hpp](#)

## 7.36 LM2 Class Reference

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

Inheritance diagram for LM2:





- $$y = A * x$$



## 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]`

$y = A \cdot x$

Implements [MultOp](#).

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

Implements [DPMethod](#).

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

Implements [DPMethod](#).

### 7.36.3 Member Data Documentation

7.36.3.1 `ldouble* LM2::f` `[private]`

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

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

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

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

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

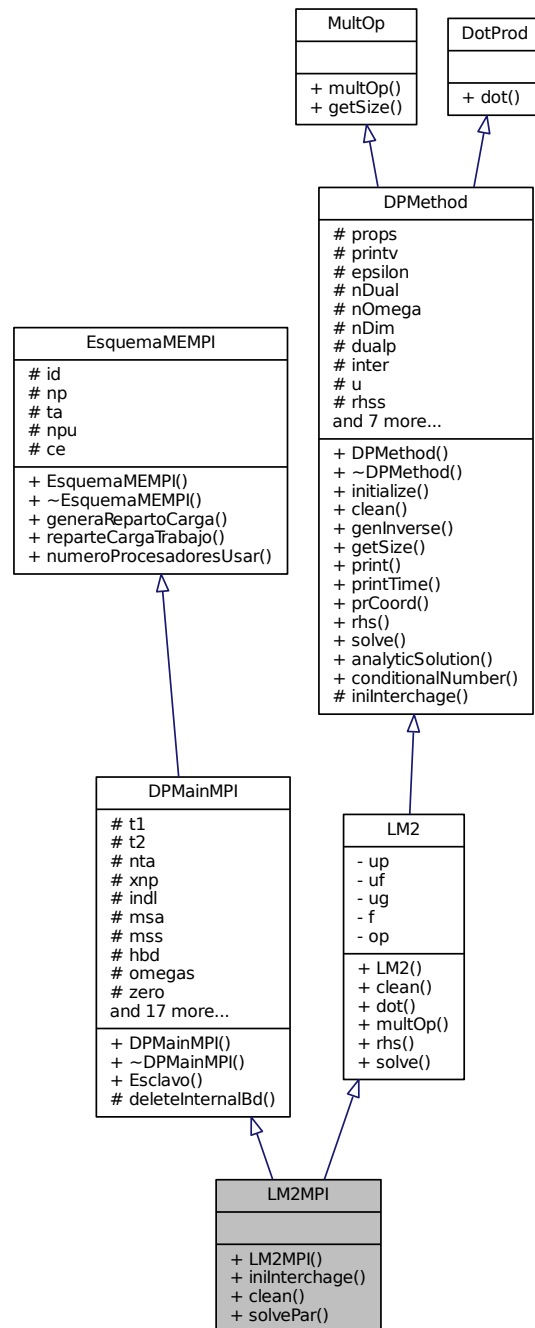
- [LM2.hpp](#)
- [LM2.cpp](#)

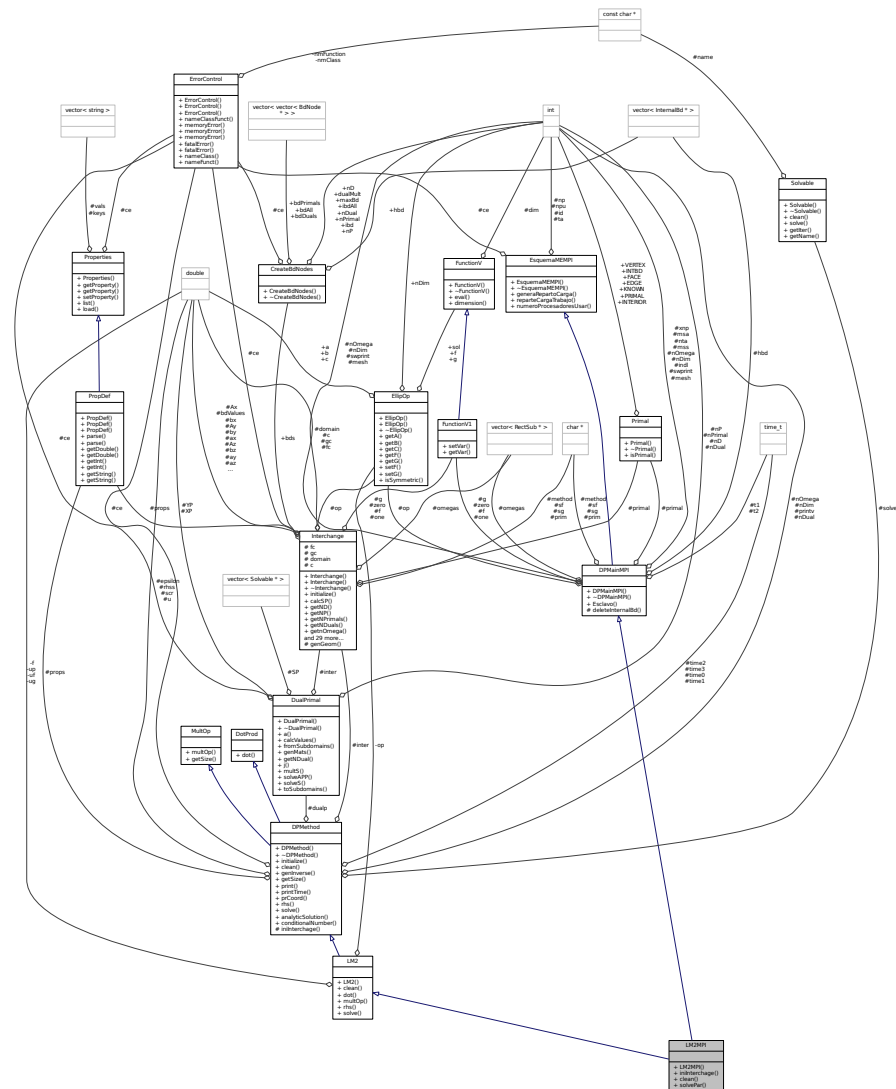
## 7.37 LM2MPI Class Reference

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

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

Inheritance diagram for LM2MPI:





## Public Member Functions

- **LM2MPI** (int **id**, int **np**, **PropDef** &**props**, **EllipOp** &**op**)  
*Constructor de la clase.*
- void **iniInterchange** (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 [DPMMethod](#).

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

Sobrecarga del la aplicacion.

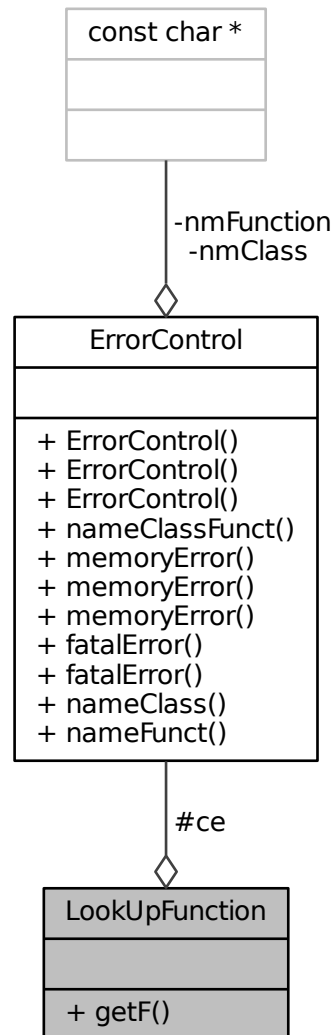
The documentation for this class was generated from the following file:

- [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:

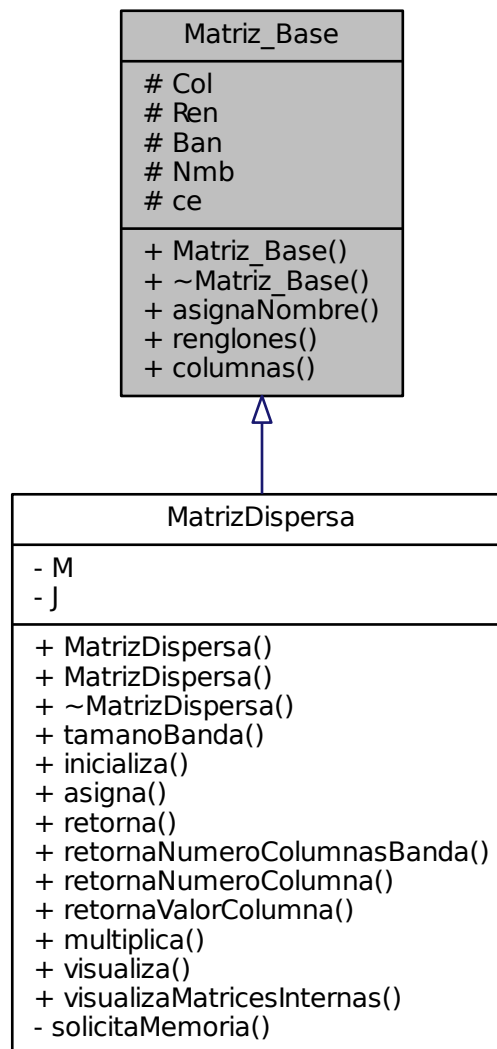
- [LookUpFunction.hpp](#)
- [LookUpFunction.cpp](#)

## 7.39 Matriz\_Base Class Reference

Clase base para el trabajar con matrices.

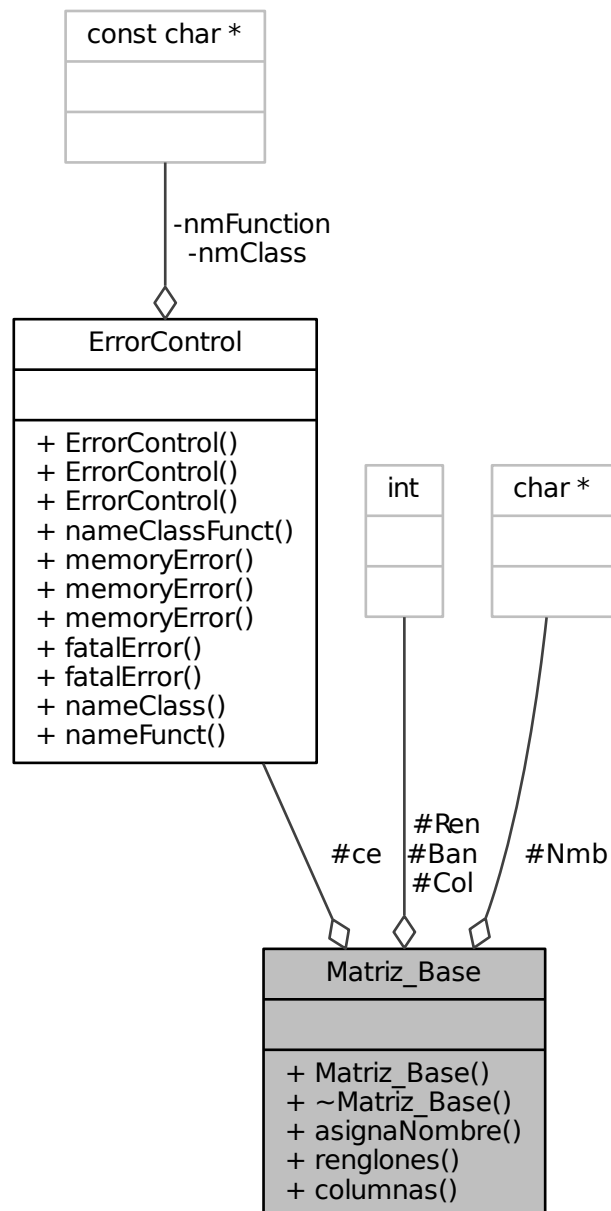
```
#include <Matriz_Base.hpp>
```

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.

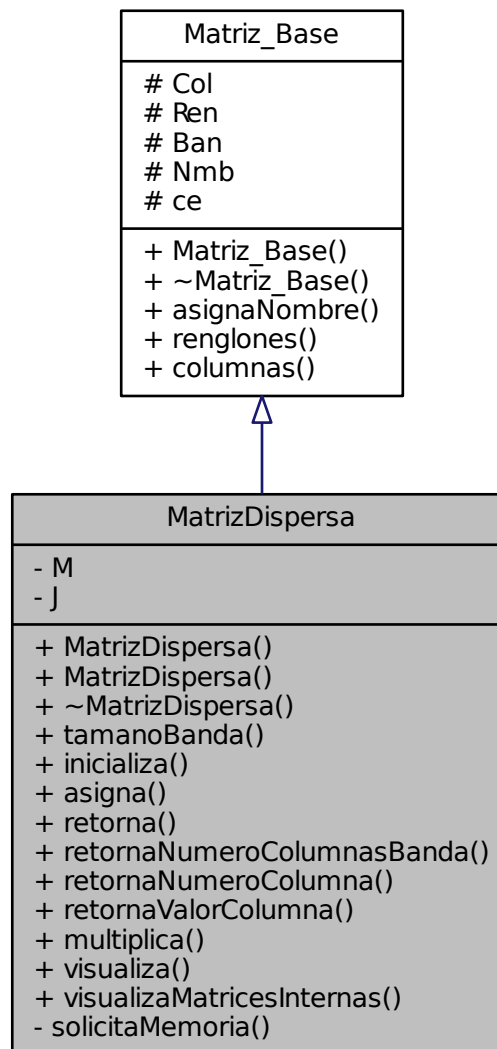
The documentation for this class was generated from the following file:

- [Matriz\\_Base.hpp](#)

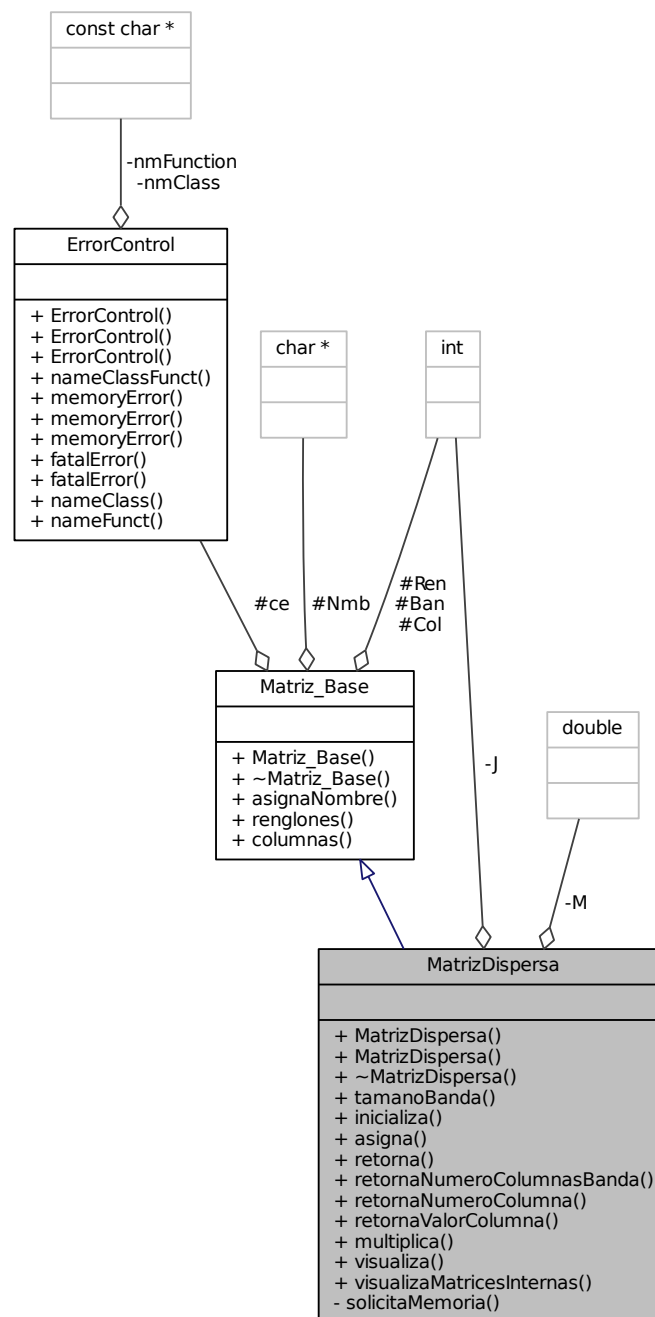
## 7.40 MatrizDispersa Class Reference

```
#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 double *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.

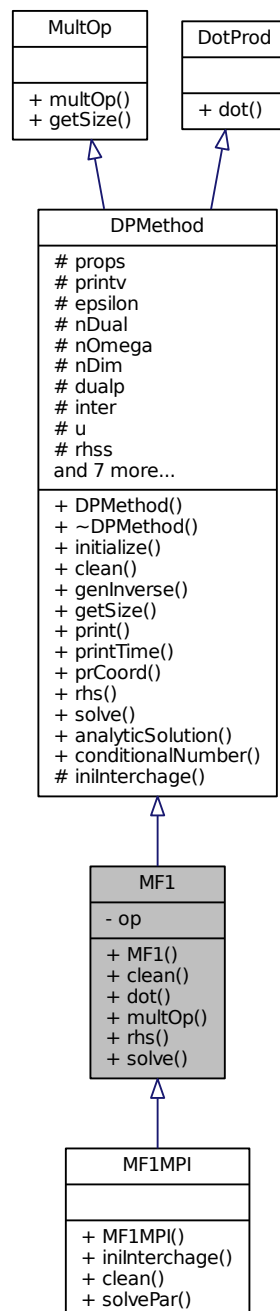
The documentation for this class was generated from the following files:

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

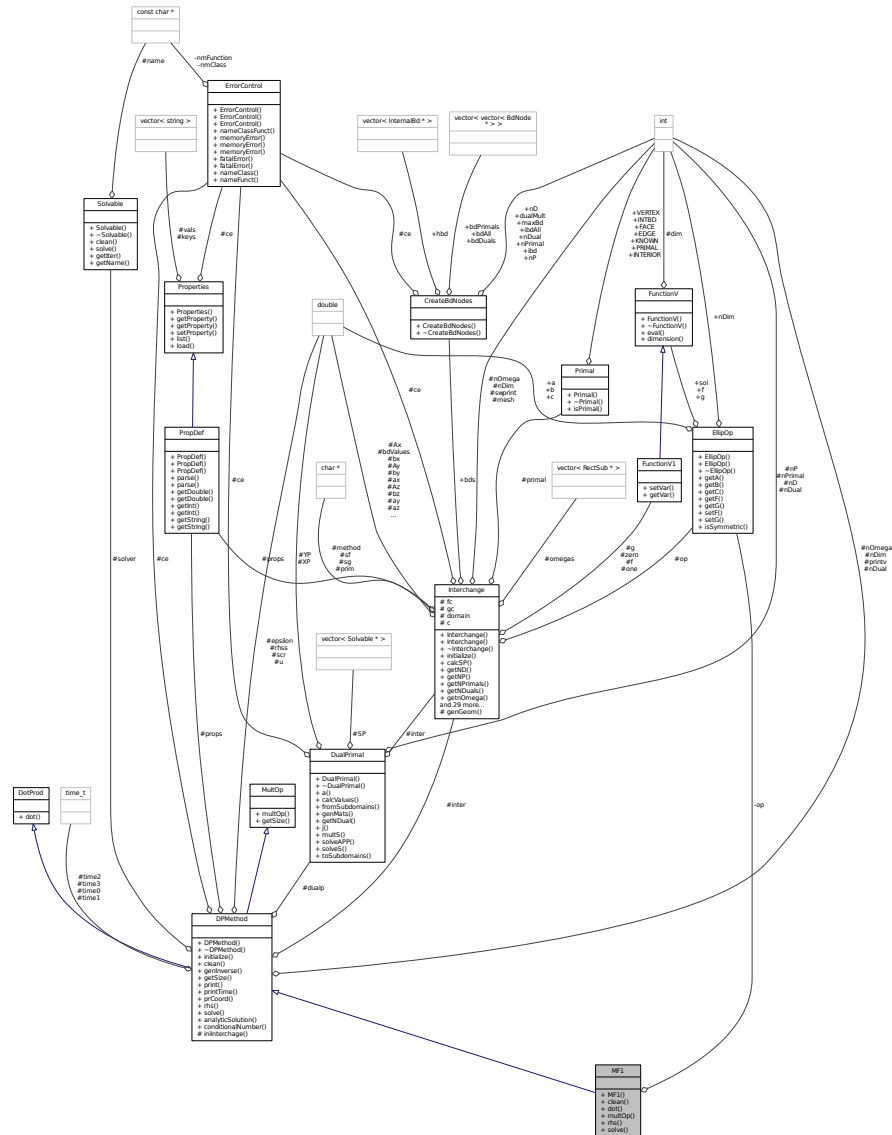
### 7.41 MF1 Class Reference

```
#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 `Idouble MF1::dot ( Idouble * u, Idouble * v )` `[virtual]`

Implements [DotProd](#).

7.41.2.3 `void MF1::multOp ( Idouble * x, Idouble * 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:

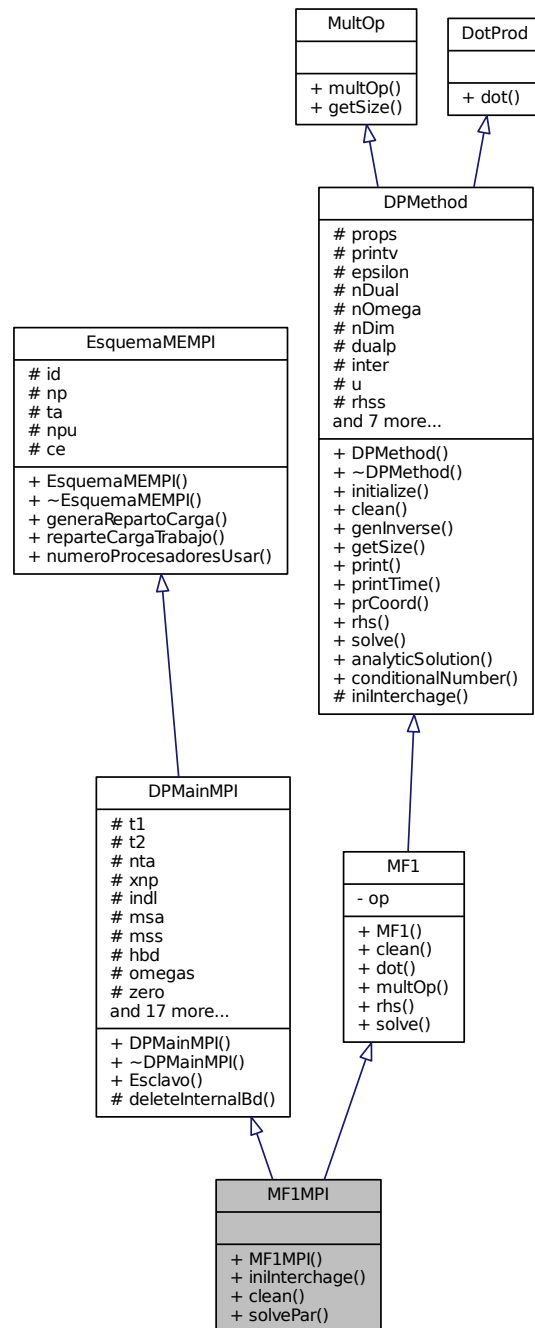
- [MF1.hpp](#)
- [MF1.cpp](#)

## 7.42 MF1MPI Class Reference

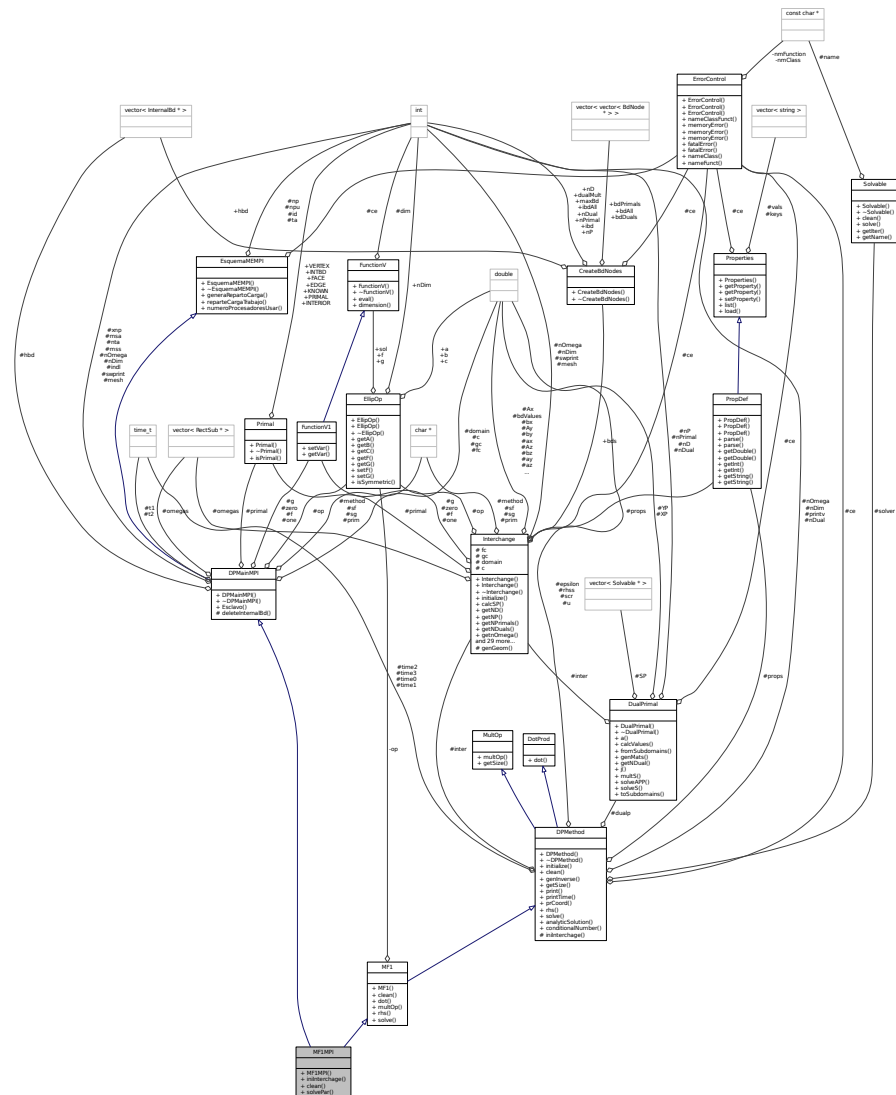
Clase para definir el metodo MF-1 de DVS-DDM.

```
#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 **iniInterchange** (void)  
*Inicializa **InterchangeMPI** en lugar de **Interchange**.*
- void **clean** (void)
- void **solvePar** (void)  
*Sobrecarga de 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.

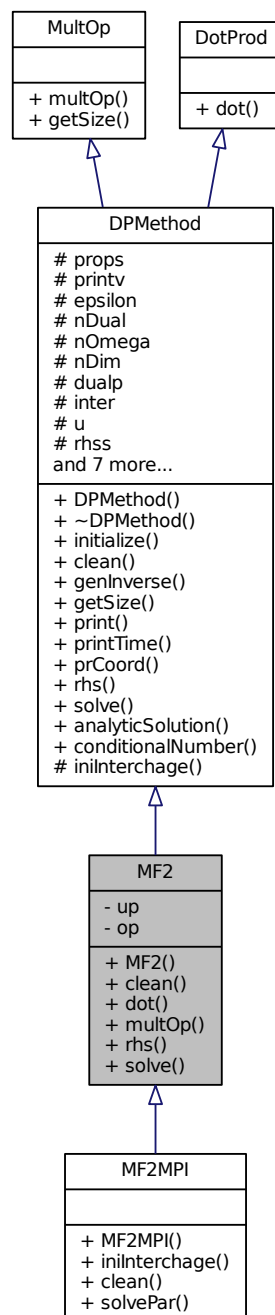
The documentation for this class was generated from the following file:

- [MF1MPI.hpp](#)

## 7.43 MF2 Class Reference

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

Inheritance diagram for MF2:





- $$y = A * x$$

## 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 )` [\[virtual\]](#)

$y = A \cdot x$

Implements [MultOp](#).

7.43.2.4 `void MF2::rhs ( void )` [\[virtual\]](#)

Implements [DPMethod](#).

7.43.2.5 `void MF2::solve ( void )` [\[virtual\]](#)

Implements [DPMethod](#).

### 7.43.3 Member Data Documentation

7.43.3.1 `EllipOp* MF2::op` [\[private\]](#)

7.43.3.2 `ldouble* MF2::up` [\[private\]](#)

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

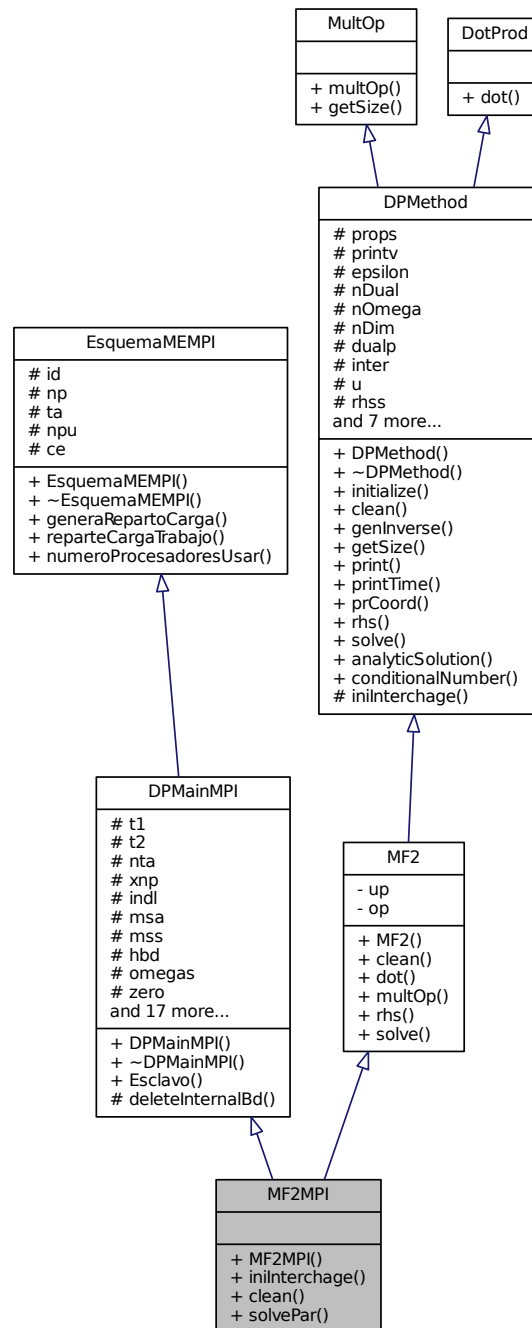
- [MF2.hpp](#)
- [MF2.cpp](#)

## 7.44 MF2MPI Class Reference

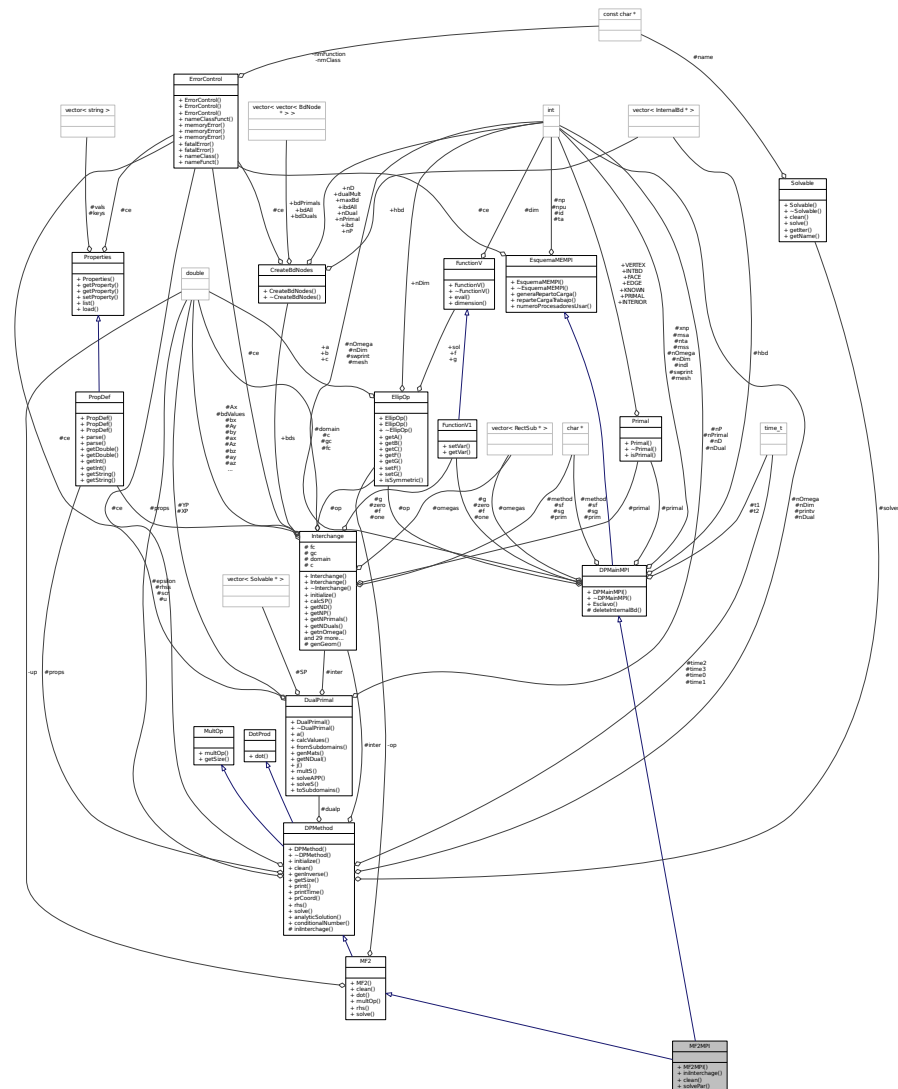
Clase para definir el metodo MF-2 de DVS-DDM.

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

Inheritance diagram for MF2MPI:



Collaboration diagram for MF2MPI:



## Public Member Functions

- **MF2MPI** (int id, int np, PropDef &props, EllipOp &op)  
*Constructor de la clase.*
- void **iniInterchange** (void)  
*Inicializa **InterchangeMPI** en lugar de **Interchange**.*
- void **clean** (void)
- void **solvePar** (void)  
*Sobrecarga de 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 [DPMMethod](#).

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

Sobrecarga del la aplicacion.

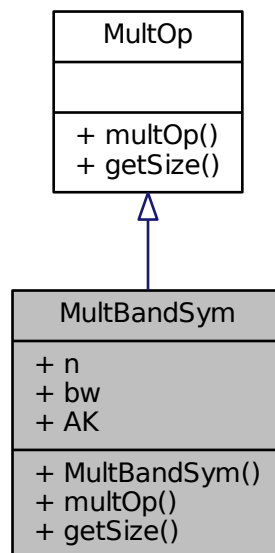
The documentation for this class was generated from the following file:

- [MF2MPI.hpp](#)

## 7.45 MultBandSym Class Reference

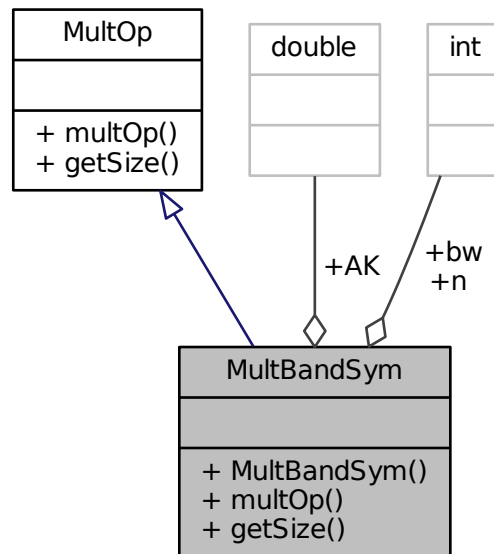
```
#include <MultBandSym.hpp>
```

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`

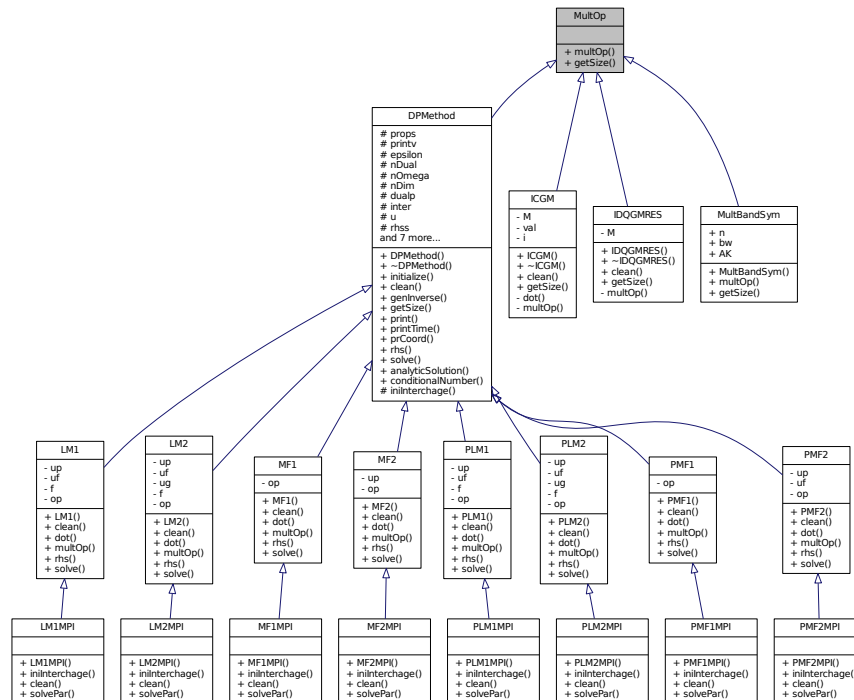
The documentation for this class was generated from the following files:

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

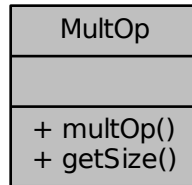
## 7.46 MultOp Class Reference

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

Inheritance diagram for MultOp:



Collaboration diagram for MultOp:



## Public Member Functions

- virtual void [multOp](#) ([Idouble](#) \*x, [Idouble](#) \*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](#), [ICGM](#), [PMF1](#), [MF1](#), [IDQGMRES](#), and [MultBandSym](#).

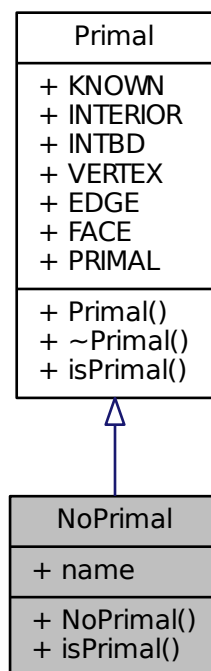
The documentation for this class was generated from the following file:

- [MultOp.hpp](#)

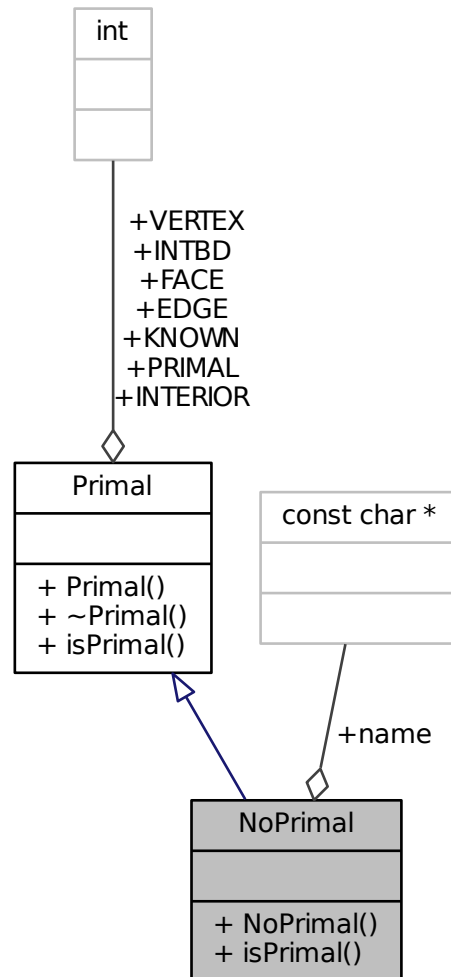
## 7.47 NoPrimal Class Reference

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

Inheritance diagram for NoPrimal:



Collaboration diagram for NoPrimal:



### Public Member Functions

- `NoPrimal` (void)
- bool `isPrimal` (int type, int \*coordN, int \*coordM)

### Public Attributes

- const char \* `name`

### Additional Inherited Members

### 7.47.1 Constructor & Destructor Documentation

7.47.1.1 `NoPrimal::NoPrimal ( void )` `[inline]`

### 7.47.2 Member Function Documentation

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

Implements [Primal](#).

### 7.47.3 Member Data Documentation

7.47.3.1 `const char* NoPrimal::name`

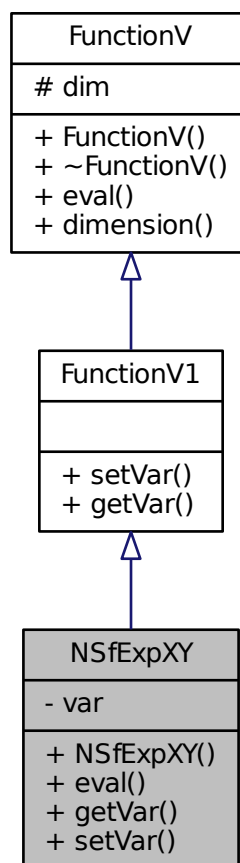
The documentation for this class was generated from the following file:

- [NoPrimal.hpp](#)

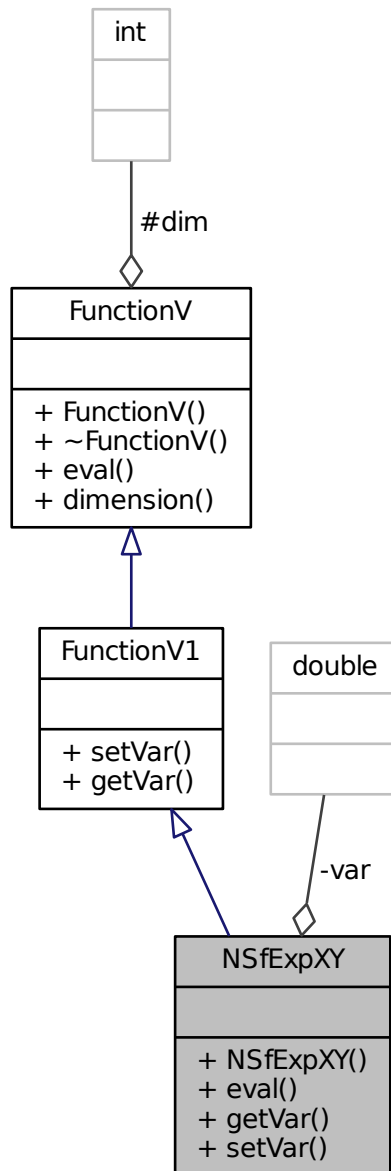
## 7.48 NSfExpXY Class Reference

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

Inheritance diagram for NSfExpXY:



Collaboration diagram for NSfExpXY:



### Public Member Functions

- `NSfExpXY` (double b)
- double `eval` (int d, double \*x)
- double `getVar` (void)
- void `setVar` (double b)



## Private Attributes

- double [var](#)

## Additional Inherited Members

### 7.48.1 Constructor & Destructor Documentation

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

### 7.48.2 Member Function Documentation

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

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

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

Implements [FunctionV1](#).

### 7.48.3 Member Data Documentation

7.48.3.1 `double NSfExpXY::var` `[private]`

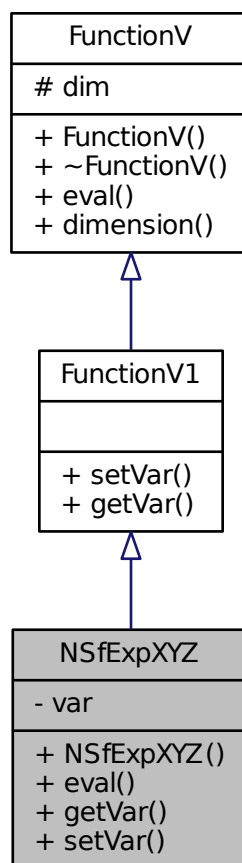
The documentation for this class was generated from the following file:

- [NSfExpXY.hpp](#)

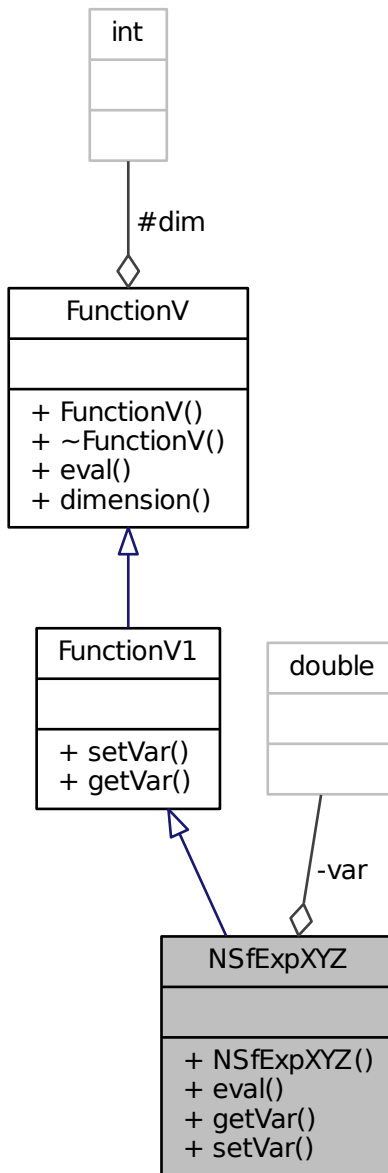
## 7.49 NSfExpXYZ Class Reference

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

Inheritance diagram for NSfExpXYZ:



Collaboration diagram for NSfExpXYZ:



## Public Member Functions

- [NSfExpXYZ](#) ([ldouble](#) b)
- [ldouble eval](#) (int d, [ldouble](#) \*x)
- [ldouble getVar](#) (void)
- void [setVar](#) ([ldouble](#) b)

## Private Attributes

- [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]`

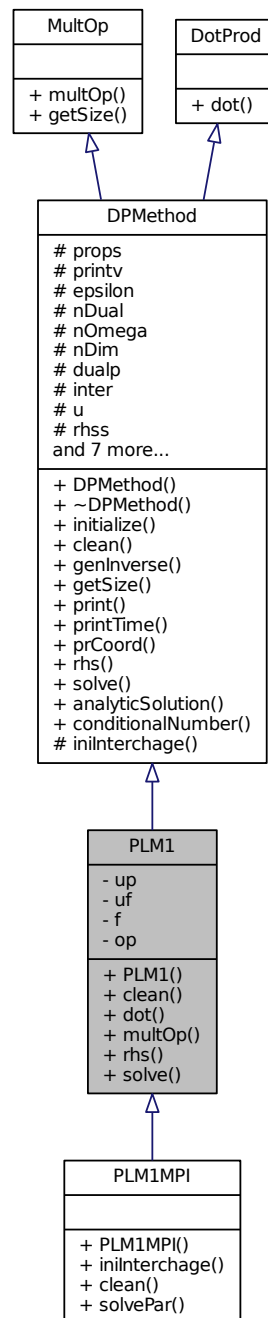
The documentation for this class was generated from the following file:

- [NSfExpXYZ.hpp](#)

## 7.50 PLM1 Class Reference

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

Inheritance diagram for PLM1:





## 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]`

$y = A \cdot x$

Implements [MultOp](#).

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

Implements [DPMethod](#).

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

Implements [DPMethod](#).

### 7.50.3 Member Data Documentation

7.50.3.1 `ldouble* PLM1::f` `[private]`

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

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

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

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

- [PLM1.hpp](#)

- [PLM1.cpp](#)

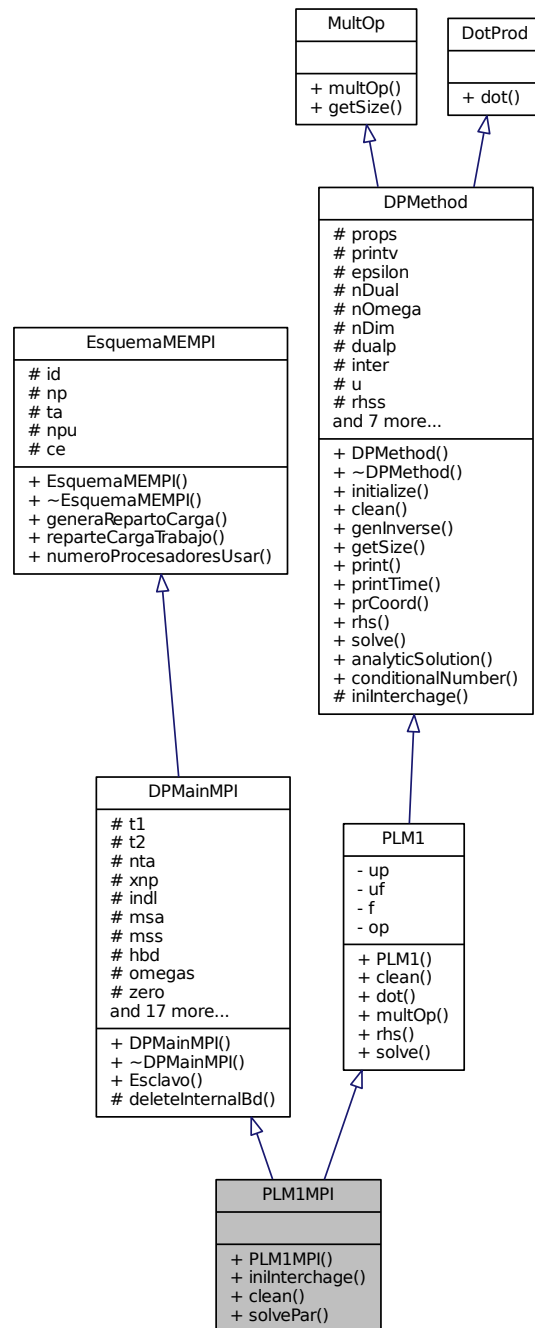
## 7.51 PLM1MPI Class Reference

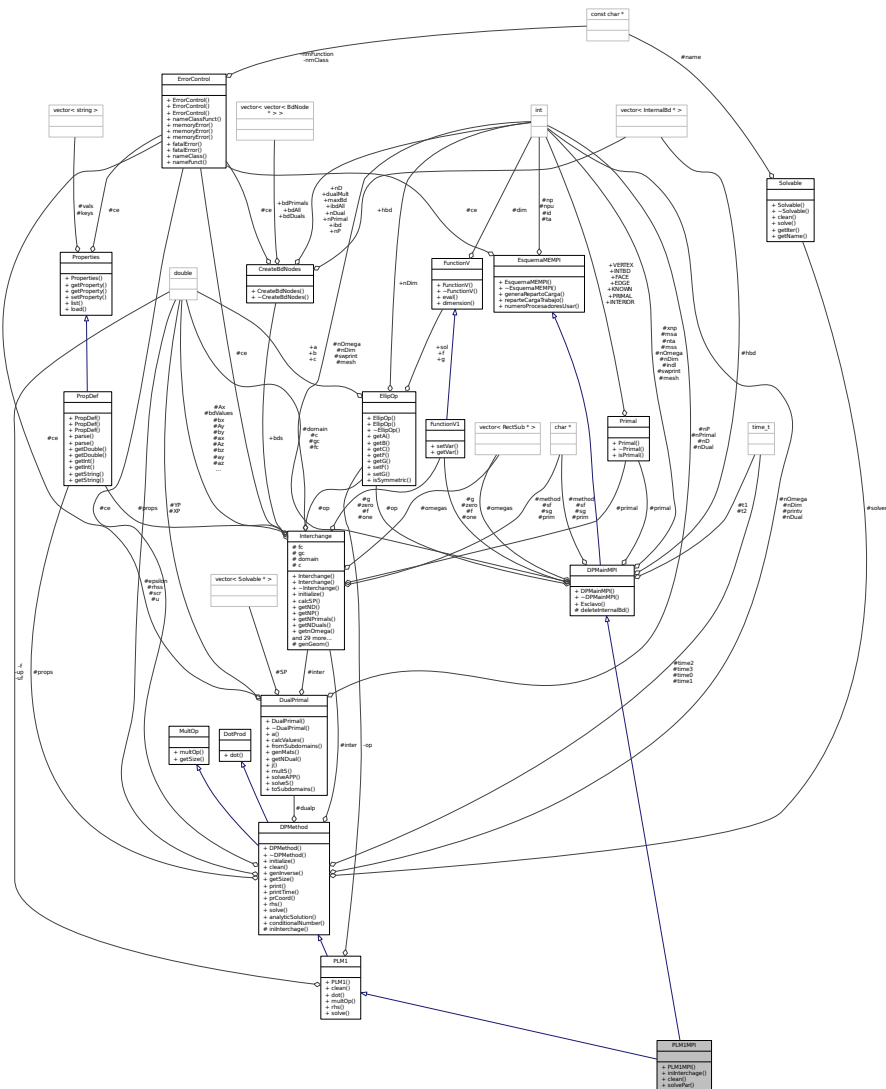
Clase para definir el metodo PLM-1 de DVS-DDM.

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



Inheritance diagram for PLM1MPI:





## Public Member Functions

- **PLM1MPI** (int **id**, int **np**, **PropDef** &**props**, **EllipOp** &**op**)  
*Constructor de la clase.*
- void **iniInterchange** (void)  
*Inicializa **InterchangeMPI** en lugar de **Interchange**.*
- void **clean** (void)
- void **solvePar** (void)  
*Sobrecarga de 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.

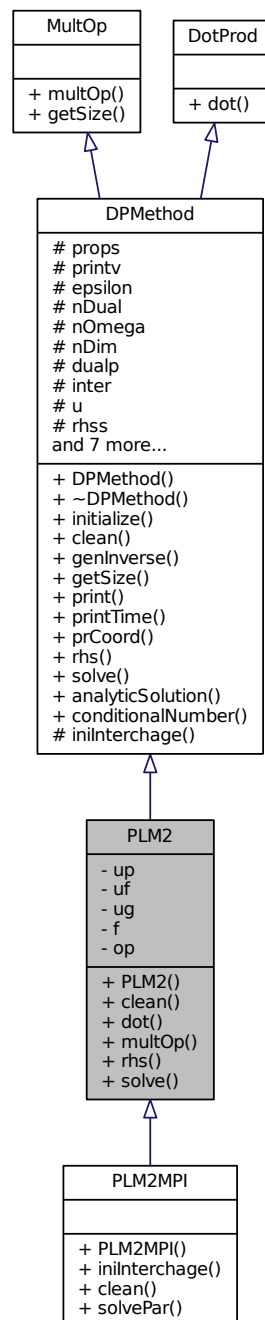
The documentation for this class was generated from the following file:

- [PLM1MPI.hpp](#)

## 7.52 PLM2 Class Reference

```
#include <PLM2.hpp>
```

Inheritance diagram for PLM2:





- $$y = A * x$$

- Generated on Tue Sep 29 2015 08:06:05 for DDM-DVS by Doxygen

## 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]`

$y = A \cdot x$

Implements [MultOp](#).

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

Implements [DPMethod](#).

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

Implements [DPMethod](#).

### 7.52.3 Member Data Documentation

7.52.3.1 `ldouble* PLM2::f` `[private]`

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

7.52.3.3 `ldouble* PLM2::uf` `[private]`

7.52.3.4 `Idouble* PLM2::ug` `[private]`

7.52.3.5 `Idouble* PLM2::up` `[private]`

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

- [PLM2.hpp](#)

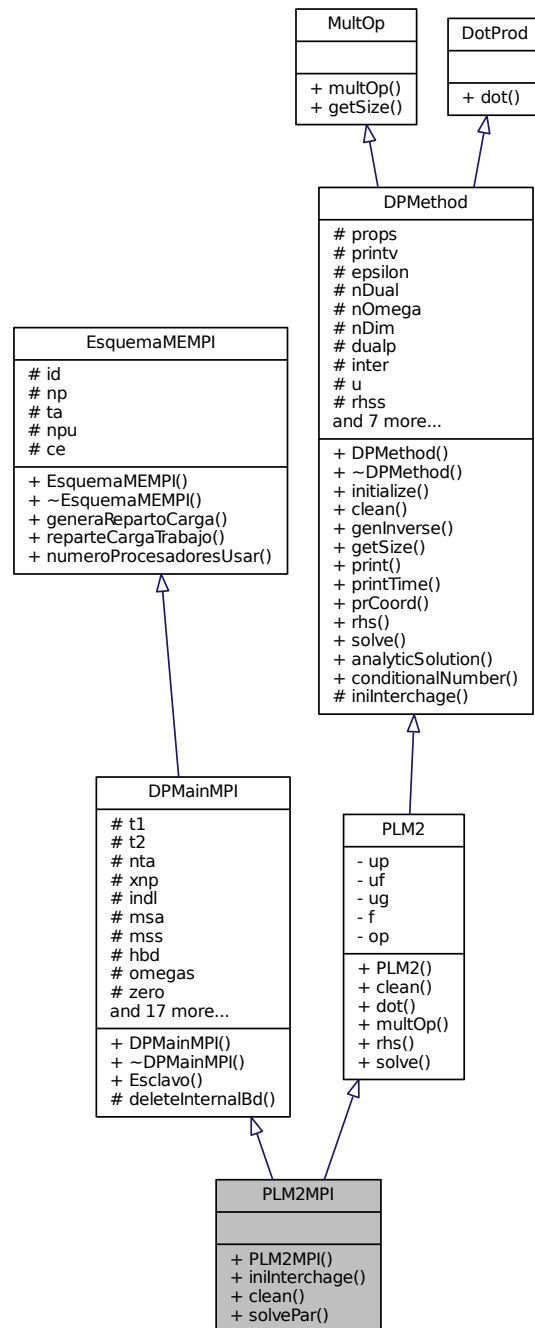
- [PLM2.cpp](#)

## 7.53 PLM2MPI Class Reference

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

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

Inheritance diagram for PLM2MPI:





[illegible]

- `PLM2MPI` (int `id`, int `np`, `PropDef` & `props`, `EllipOp` & `op`)  
*Constructor de la clase.*
- void `iniInterchange` (void)  
*Inicializa `InterchangeMPI` en lugar de `Interchange`.*
- void `clean` (void)
- void `solvePar` (void)  
*Sobrecarga del la aplicacion.*

Generated on Tue Sep 29 2015 08:06:05 for DDM-DVS by Doxygen

### 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 [DPMMethod](#).

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

Sobrecarga del la aplicacion.

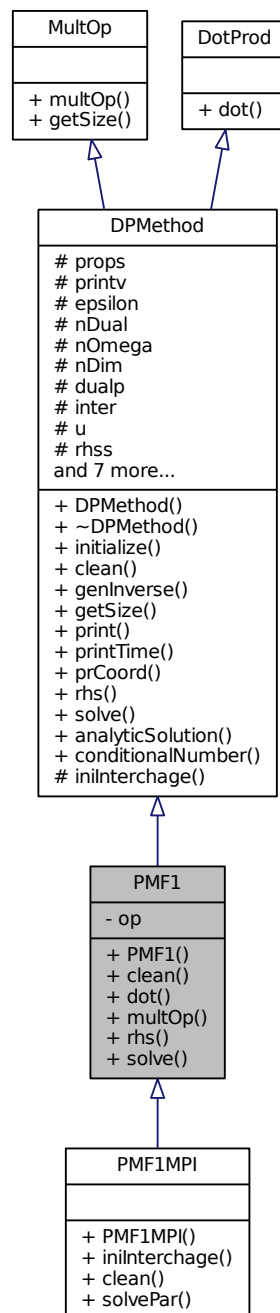
The documentation for this class was generated from the following file:

- [PLM2MPI.hpp](#)

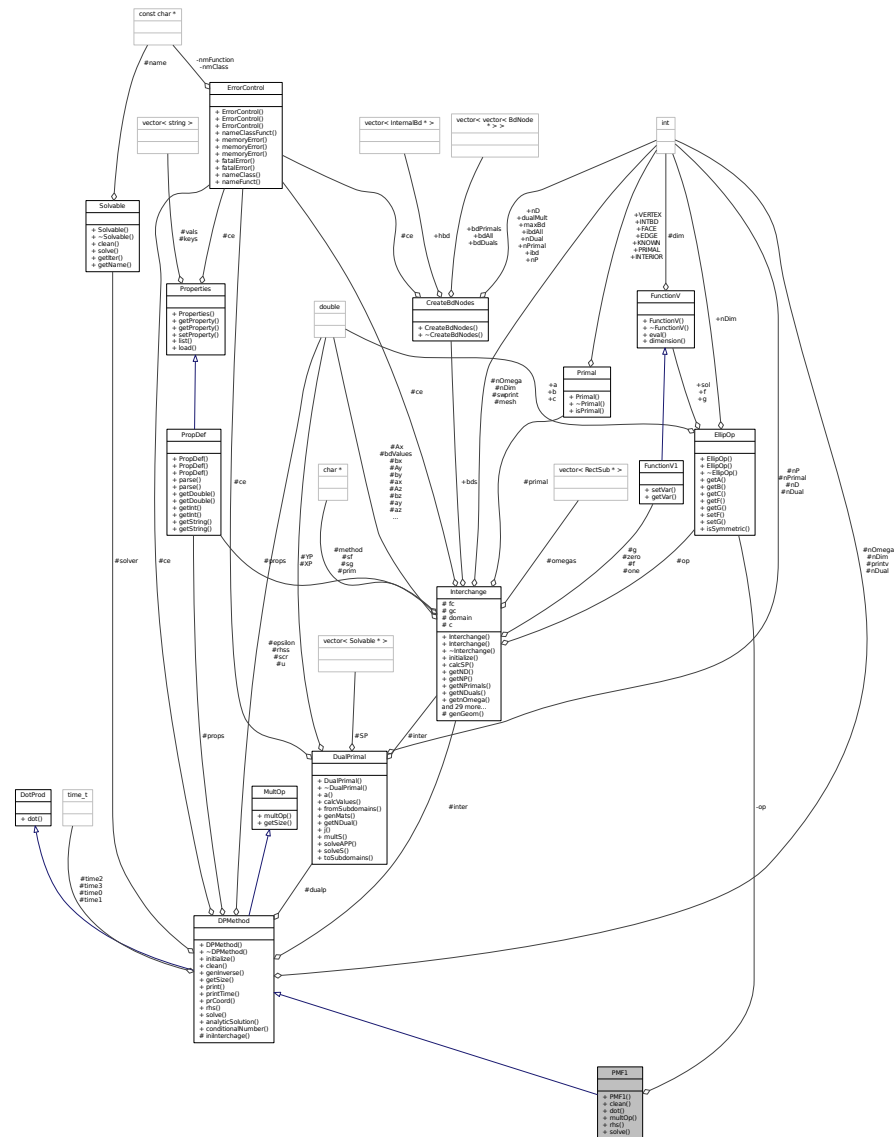
## 7.54 PMF1 Class Reference

```
#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]`

$y = A \cdot x$

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:

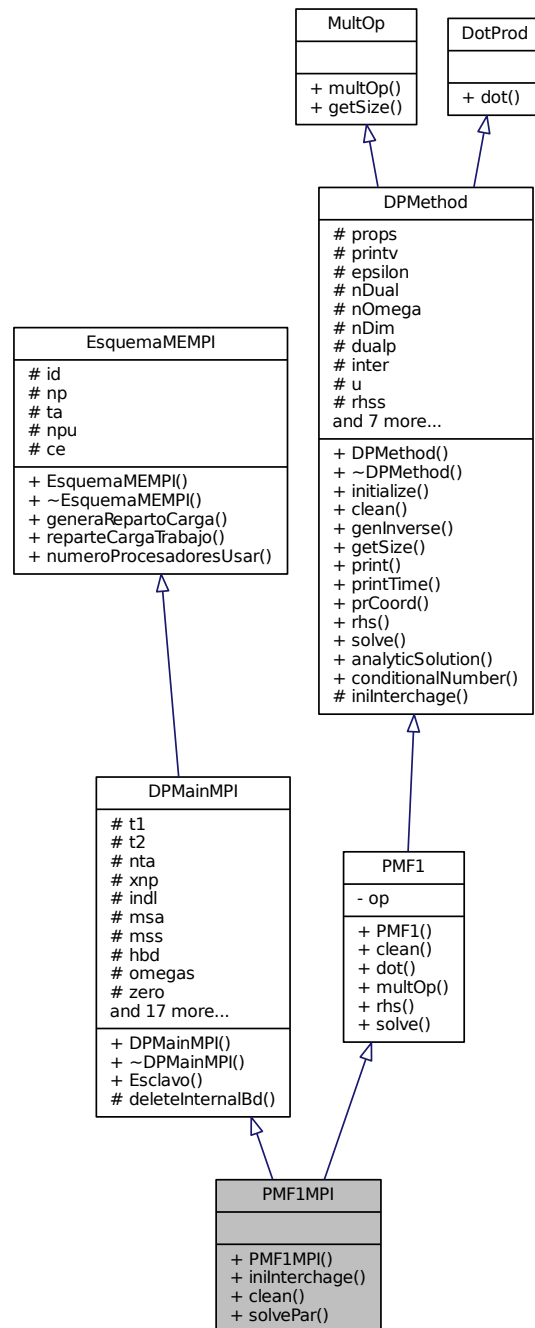
- [PMF1.hpp](#)
- [PMF1.cpp](#)

## 7.55 PMF1MPI Class Reference

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

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

Inheritance diagram for PMF1MPI:



[illegible]

- **PMF1MPI** (int **id**, int **np**, **PropDef** &**props**, **EllipOp** &**op**)  
*Constructor de la clase.*
- void **iniInterchange** (void)  
*Inicializa **InterchangeMPI** en lugar de **Interchange**.*
- void **clean** (void)
- void **solvePar** (void)  
*Sobrecarga del la aplicacion.*

Generated on Tue Sep 29 2015 08:06:05 for DDM-DVS by Doxygen

### 7.55.1 Detailed Description

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

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

#### Author

Antonio Carrillo Ledesma

#### Date

primavera 2010

#### Version

1.0.0

**Bug** No hay errores conocidos

### 7.55.2 Constructor & Destructor Documentation

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

Constructor de la clase.

### 7.55.3 Member Function Documentation

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

Reimplemented from [PMF1](#).

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

Inicializa [InterchangeMPI](#) en lugar de [Interchange](#).

Reimplemented from [DPMethod](#).

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

Sobrecarga del la aplicacion.

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

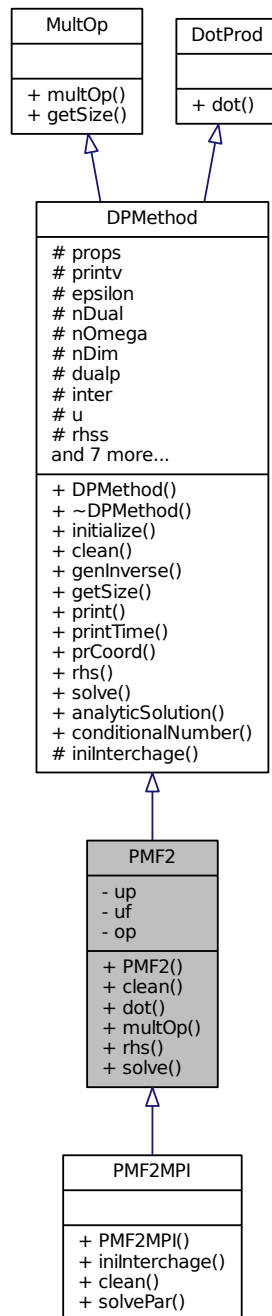
- [PMF1MPI.hpp](#)

## 7.56 PMF2 Class Reference

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



Inheritance diagram for PMF2:





- $$y = A * x$$

## 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]`

$y = A \cdot x$

Implements [MultOp](#).

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

Implements [DPMethod](#).

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

Implements [DPMethod](#).

### 7.56.3 Member Data Documentation

7.56.3.1 `EllipOp* PMF2::op` `[private]`

7.56.3.2 `ldouble* PMF2::uf` `[private]`

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

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

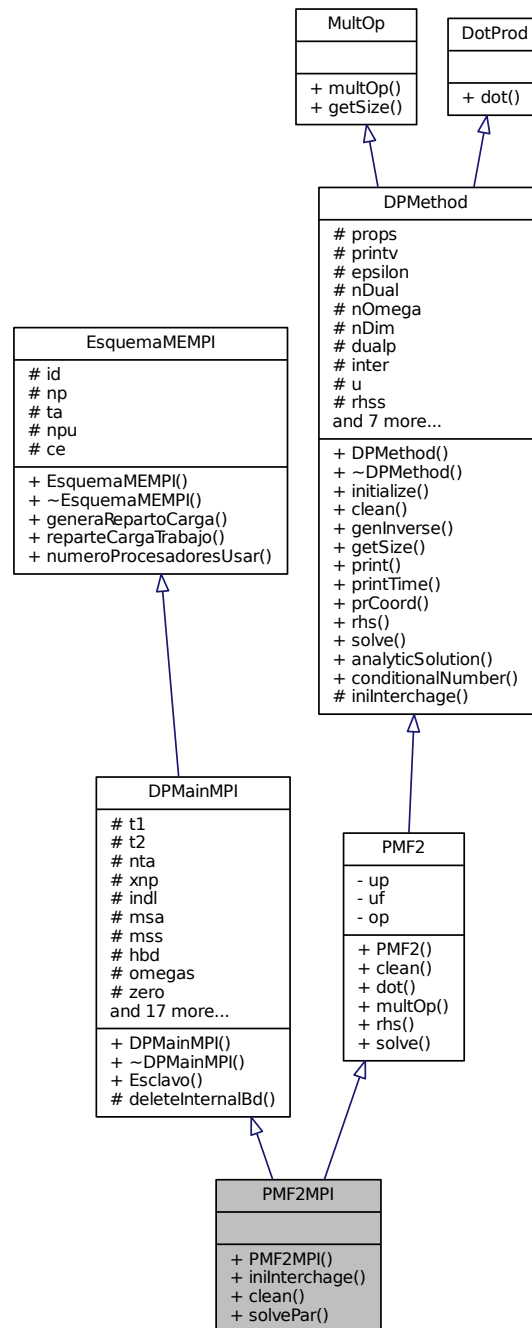
- [PMF2.hpp](#)
- [PMF2.cpp](#)

## 7.57 PMF2MPI Class Reference

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

```
#include <PMF2MPI.hpp>
```

Inheritance diagram for PMF2MPI:



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

- void **iniInterchage** (void)

Inicializa **InterchangeMPI** en lugar de **Interchange**.

- void solvePar (void)

Generated on Tue Sep 29 2015 08:06:05 for DDM-DVS by Doxygen

### 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.

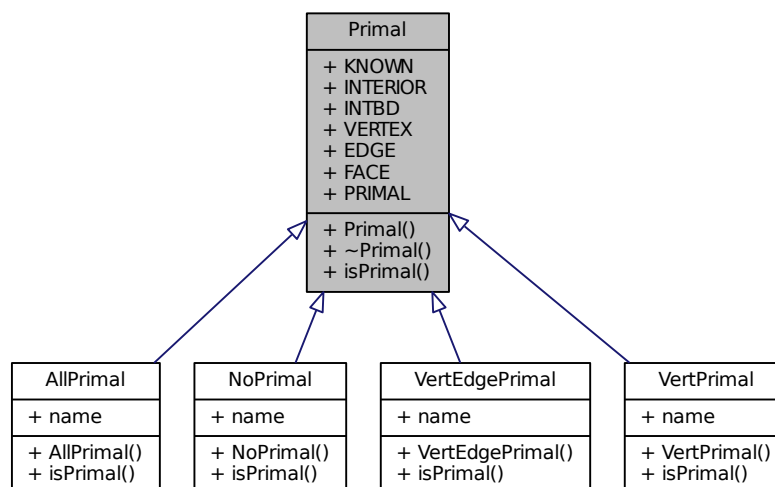
The documentation for this class was generated from the following file:

- [PMF2MPI.hpp](#)

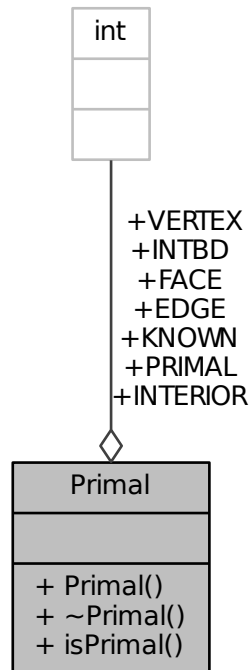
## 7.58 Primal Class Reference

```
#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]`

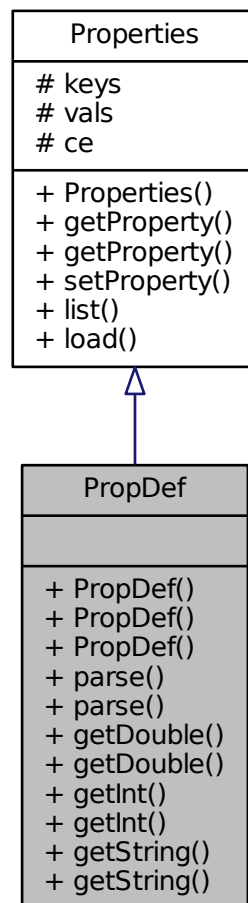
The documentation for this class was generated from the following file:

- [Primal.hpp](#)

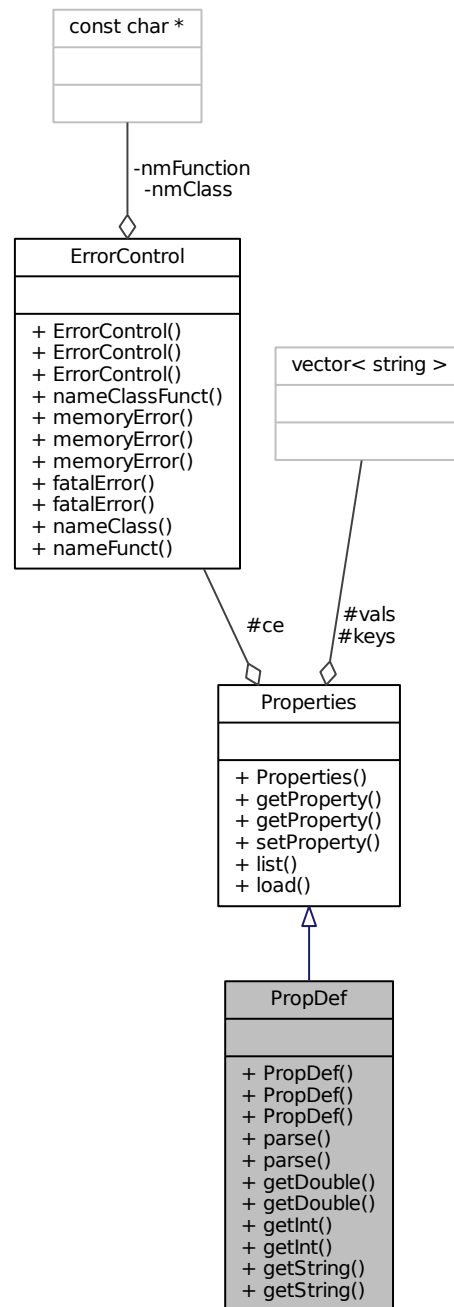
## 7.59 PropDef Class Reference

```
#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[ ])
- [Idouble](#) [getDouble](#) (const char \*key, [Idouble](#) value)
- [Idouble](#) [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 `Idouble PropDef::getDouble ( const char * key, Idouble value )`

7.59.2.2 `Idouble PropDef::getDouble ( const char * key )`

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

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

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

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

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

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

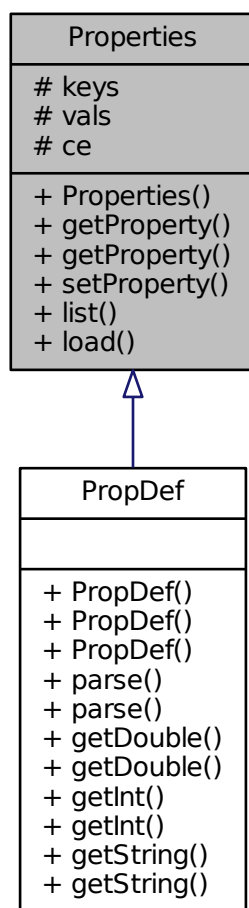
The documentation for this class was generated from the following files:

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

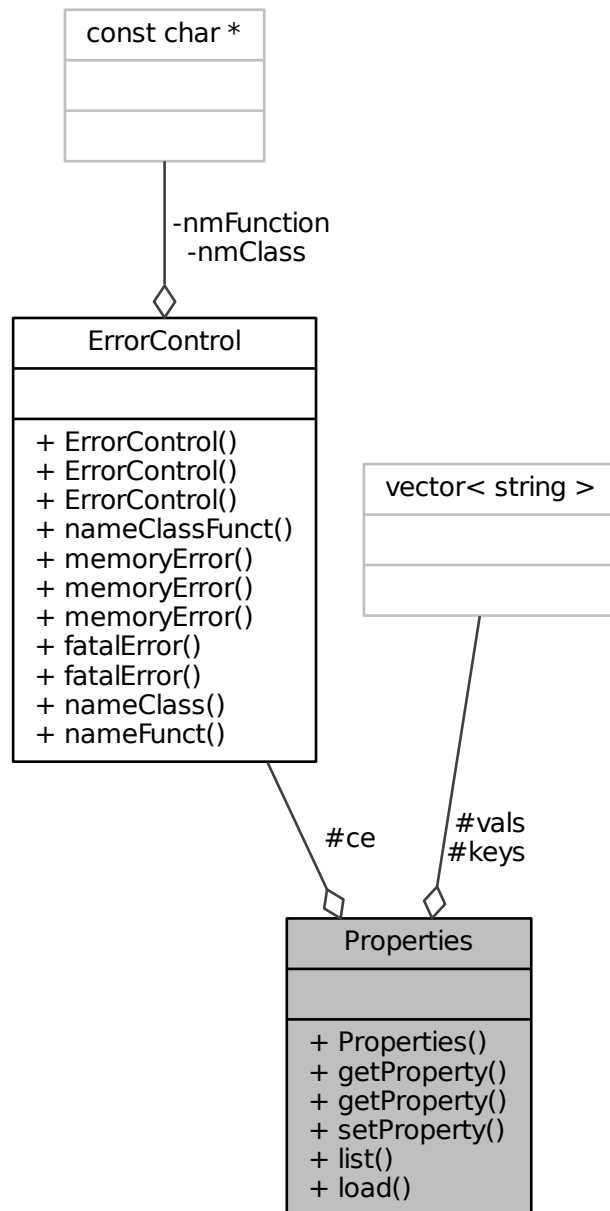
## 7.60 Properties Class Reference

```
#include <Properties.hpp>
```

Inheritance diagram for Properties:



Collaboration diagram for Properties:



## Public Member Functions

- [Properties](#) (void)
- char \* [getProperty](#) (const char \*s, const char \*val)
- const char \* [getProperty](#) (const char \*s)

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

### Protected Attributes

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

*Control de errores.*

### 7.60.1 Constructor & Destructor Documentation

7.60.1.1 `Properties::Properties ( void )` `[inline]`

### 7.60.2 Member Function Documentation

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

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

7.60.2.3 `void Properties::list ( void )`

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

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

### 7.60.3 Member Data Documentation

7.60.3.1 `ErrorControl Properties::ce` `[protected]`

Control de errores.

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

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

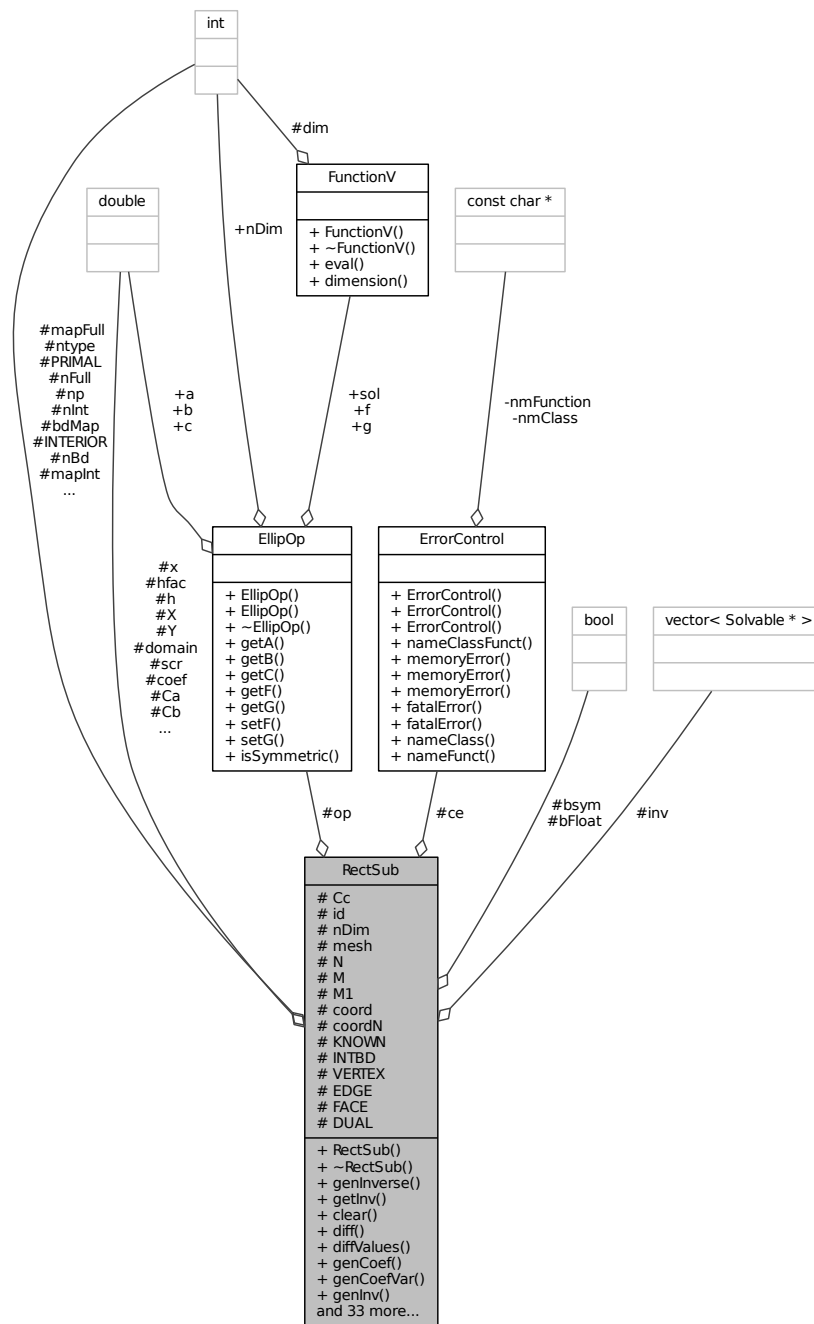
The documentation for this class was generated from the following files:

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

## 7.61 RectSub Class Reference

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

Collaboration diagram for RectSub:



## Public Member Functions

- **RectSub** (int `id`, int `nDim`, int `*mesh`, `ldouble **dom`, **EllipOp** `&op`, **Primal** `&primal`)
- **~RectSub** (void)



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

### Protected Attributes

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

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

*Control de errores.*

## Static Protected Attributes

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

## 7.61.1 Constructor & Destructor Documentation

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

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

## 7.61.2 Member Function Documentation

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

- 7.61.2.2 `void RectSub::diff ( int sc3, int sc1, int sc2 )`
- 7.61.2.3 `void RectSub::diffValues ( int sc, Idouble * u ) [inline]`
- 7.61.2.4 `void RectSub::genCoef ( EllipOp & op )`
- 7.61.2.5 `void RectSub::genCoefVar ( int ren )`
- 7.61.2.6 `void RectSub::genInv ( int type )`
- 7.61.2.7 `Solvable * RectSub::genInverse ( int * map, Idouble fac )`
- 7.61.2.8 `void RectSub::genNcoord ( int n, int * coord, int * N )`
- 7.61.2.9 `void RectSub::genNtype ( Primal & primal )`
- 7.61.2.10 `int RectSub::getBdSize ( ) [inline]`
- 7.61.2.11 `void RectSub::getCoord ( int m, Idouble * x )`
- 7.61.2.12 `void RectSub::getCoordNode ( int n, Idouble * x ) [inline]`
- 7.61.2.13 `vector< InternalBd * > RectSub::getInternalBd ( void )`
- 7.61.2.14 `vector< Solvable * > RectSub::getInv ( void ) [inline]`
- 7.61.2.15 `int RectSub::getNP ( void ) [inline]`
- 7.61.2.16 `int* RectSub::getNtype ( void ) [inline]`
- 7.61.2.17 `void RectSub::getPrimals ( int sc, Idouble * u )`
- 7.61.2.18 `Idouble RectSub::getValue ( int sc, int n ) [inline]`
- 7.61.2.19 `void RectSub::getValues ( int sc, Idouble * u )`
- 7.61.2.20 `void RectSub::inverse ( int sp, int sc1, int sc2 )`
- 7.61.2.21 `bool RectSub::isDual ( int i ) [inline]`
- 7.61.2.22 `bool RectSub::isFloat ( void ) [inline]`
- 7.61.2.23 `bool RectSub::isIntBd ( int * coord )`
- 7.61.2.24 `bool RectSub::isInterior ( int * coord )`
- 7.61.2.25 `bool RectSub::isInterior ( int i ) [inline]`
- 7.61.2.26 `bool RectSub::isKnown ( int * coord )`
- 7.61.2.27 `bool RectSub::isKnown ( int i ) [inline]`

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

### 7.61.3 Member Data Documentation

- 7.61.3.1 `int* RectSub::bdMap` `[protected]`
- 7.61.3.2 `bool RectSub::bFloat` `[protected]`
- 7.61.3.3 `bool RectSub::bsym` `[protected]`
- 7.61.3.4 `Idouble RectSub::Ca[3]` `[protected]`
- 7.61.3.5 `Idouble RectSub::Cb[3]` `[protected]`
- 7.61.3.6 `Idouble RectSub::Cc` `[protected]`
- 7.61.3.7 `ErrorControl RectSub::ce` `[protected]`

Control de errores.

- 7.61.3.8 `Idouble* RectSub::coef` `[protected]`
- 7.61.3.9 `int* RectSub::coord` `[protected]`
- 7.61.3.10 `int* RectSub::coordN` `[protected]`

- 7.61.3.11 `Idouble** RectSub::domain` [protected]
- 7.61.3.12 `const int RectSub::DUAL = 128` [static], [protected]
- 7.61.3.13 `const int RectSub::EDGE = 16` [static], [protected]
- 7.61.3.14 `const int RectSub::FACE = 32` [static], [protected]
- 7.61.3.15 `Idouble* RectSub::h` [protected]
- 7.61.3.16 `Idouble RectSub::hfac` [protected]
- 7.61.3.17 `int RectSub::id` [protected]
- 7.61.3.18 `const int RectSub::INTBD = 4` [static], [protected]
- 7.61.3.19 `const int RectSub::INTERIOR = 2` [static], [protected]
- 7.61.3.20 `vector<Solvable*> RectSub::inv` [protected]
- 7.61.3.21 `const int RectSub::KNOWN = 1` [static], [protected]
- 7.61.3.22 `int* RectSub::M` [protected]
- 7.61.3.23 `int* RectSub::M1` [protected]
- 7.61.3.24 `int* RectSub::mapFull` [protected]
- 7.61.3.25 `int* RectSub::mapInt` [protected]
- 7.61.3.26 `int* RectSub::mesh` [protected]
- 7.61.3.27 `int* RectSub::N` [protected]
- 7.61.3.28 `int RectSub::nBd` [protected]
- 7.61.3.29 `int RectSub::nDim` [protected]
- 7.61.3.30 `int RectSub::nFull` [protected]
- 7.61.3.31 `int RectSub::nInt` [protected]
- 7.61.3.32 `int RectSub::np` [protected]
- 7.61.3.33 `int* RectSub::ntype` [protected]
- 7.61.3.34 `EllipOp* RectSub::op` [protected]
- 7.61.3.35 `const int RectSub::PRIMAL = 64` [static], [protected]
- 7.61.3.36 `Idouble** RectSub::scr` [protected]

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

7.61.3.38 `Idouble* RectSub::X` `[protected]`

7.61.3.39 `Idouble* RectSub::x` `[protected]`

7.61.3.40 `Idouble* RectSub::Y` `[protected]`

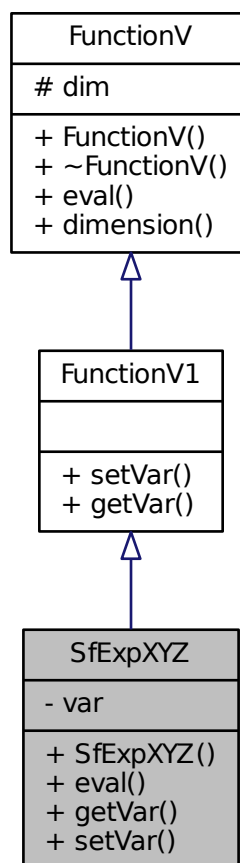
The documentation for this class was generated from the following files:

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

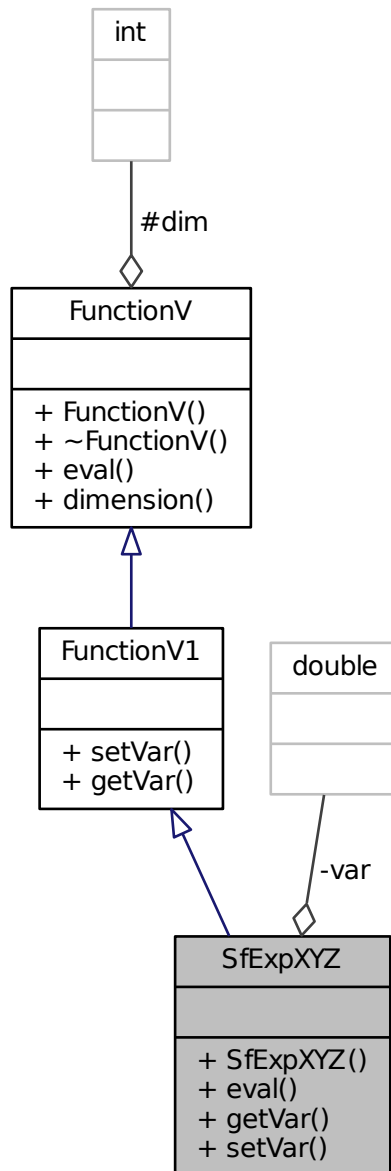
## 7.62 SfExpXYZ Class Reference

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

Inheritance diagram for SfExpXYZ:



Collaboration diagram for SfExpXYZ:



### Public Member Functions

- `SfExpXYZ` (`ldouble` b)
- `ldouble eval` (`int` d, `ldouble` \*x)
- `ldouble getVar` (`void`)
- `void setVar` (`ldouble` b)



## Private Attributes

- [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]`

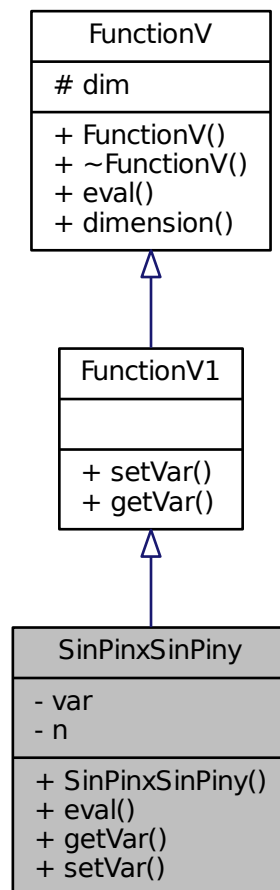
The documentation for this class was generated from the following file:

- [SfExpXYZ.hpp](#)

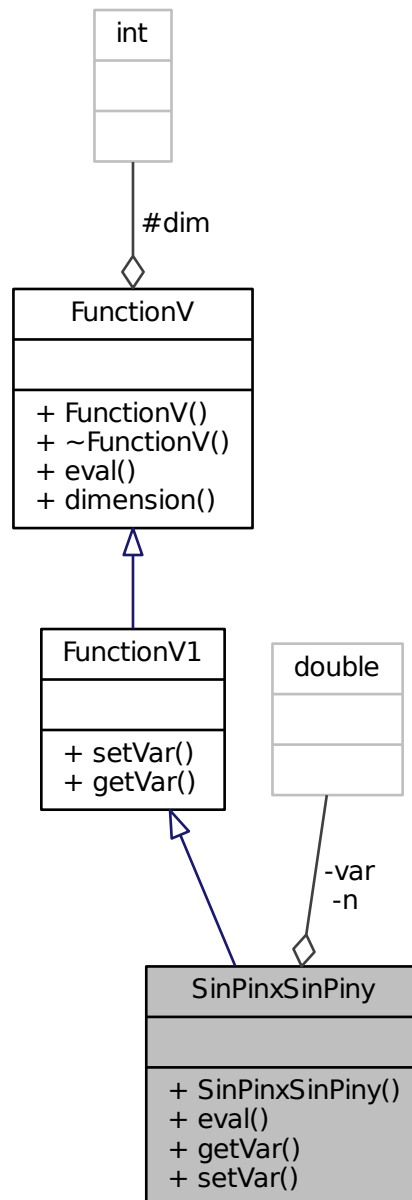
## 7.63 SinPinxSinPiny Class Reference

```
#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](#) ([Idouble](#) b)

### Private Attributes

- [Idouble](#) var
- [Idouble](#) n

### Additional Inherited Members

#### 7.63.1 Constructor & Destructor Documentation

7.63.1.1 [SinPinxSinPiny::SinPinxSinPiny](#) ( [Idouble](#) *b* ) `[inline]`

#### 7.63.2 Member Function Documentation

7.63.2.1 [Idouble](#) [SinPinxSinPiny::eval](#) ( `int` *d*, [Idouble](#) \* *x* ) `[inline]`, `[virtual]`

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

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

Implements [FunctionV1](#).

#### 7.63.3 Member Data Documentation

7.63.3.1 [Idouble](#) [SinPinxSinPiny::n](#) `[private]`

7.63.3.2 [Idouble](#) [SinPinxSinPiny::var](#) `[private]`

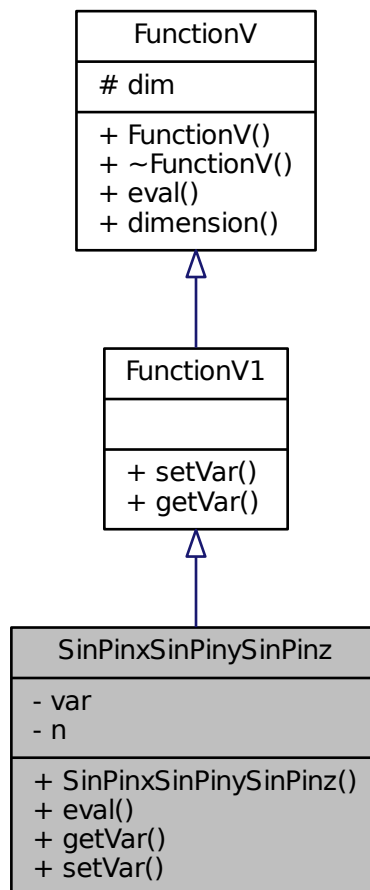
The documentation for this class was generated from the following file:

- [SinPinxSinPiny.hpp](#)

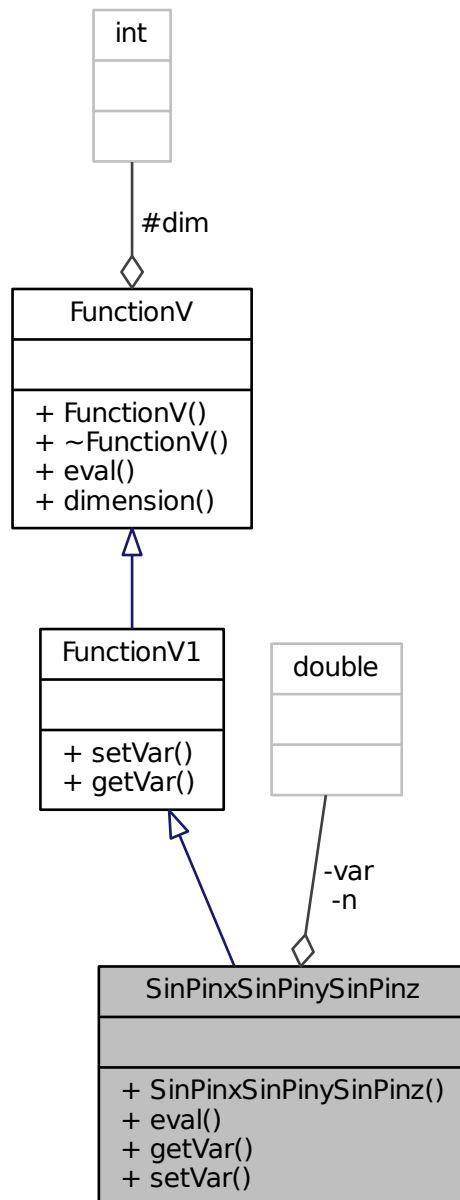
## 7.64 SinPinxSinPinySinPinz Class Reference

```
#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](#) ([Idouble](#) b)

### Private Attributes

- [Idouble](#) var
- [Idouble](#) n

### Additional Inherited Members

#### 7.64.1 Constructor & Destructor Documentation

7.64.1.1 [SinPixSinPinySinPinz::SinPixSinPinySinPinz \( \[Idouble\]\(#\) b \)](#) [\[inline\]](#)

#### 7.64.2 Member Function Documentation

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

Implements [FunctionV](#).

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

Implements [FunctionV1](#).

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

Implements [FunctionV1](#).

#### 7.64.3 Member Data Documentation

7.64.3.1 [Idouble SinPixSinPinySinPinz::n](#) [\[private\]](#)

7.64.3.2 [Idouble SinPixSinPinySinPinz::var](#) [\[private\]](#)

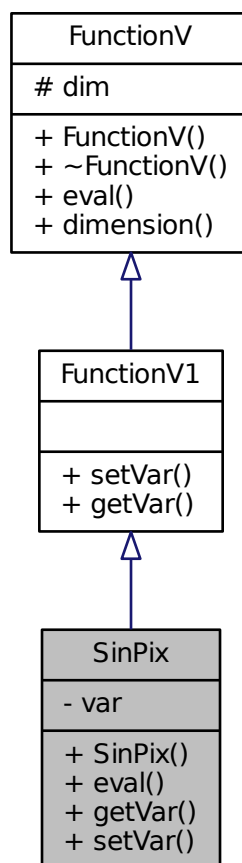
The documentation for this class was generated from the following file:

- [SinPixSinPinySinPinz.hpp](#)

## 7.65 SinPix Class Reference

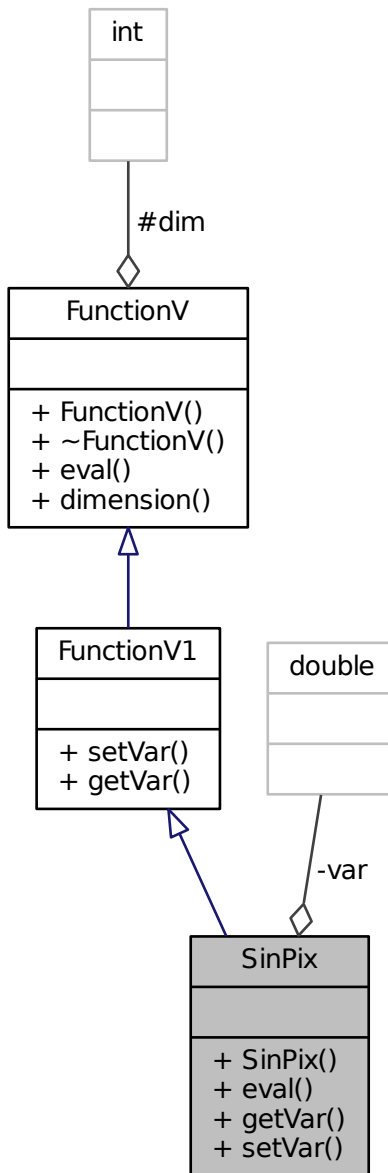
```
#include <SinPix.hpp>
```

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]`

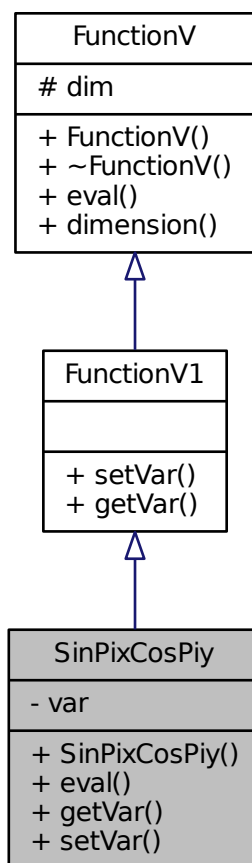
The documentation for this class was generated from the following file:

- [SinPix.hpp](#)

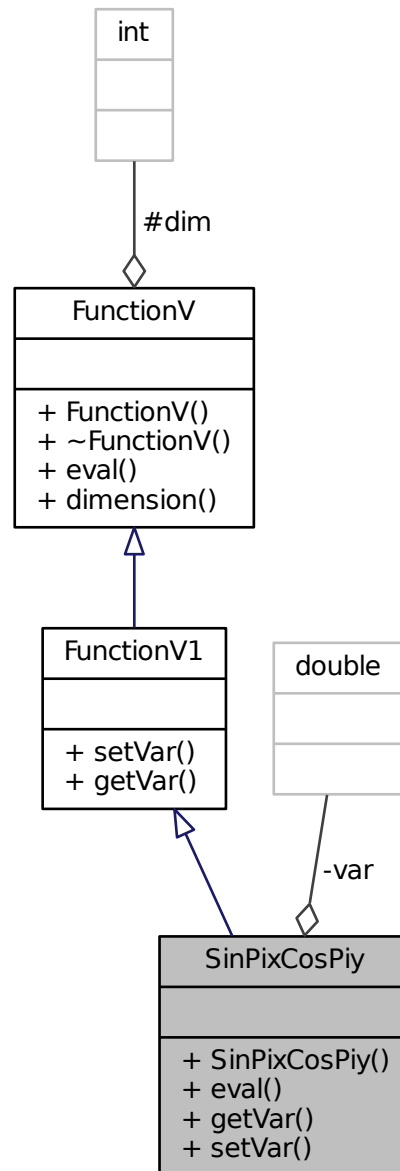
## 7.66 SinPixCosPiy Class Reference

```
#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]`

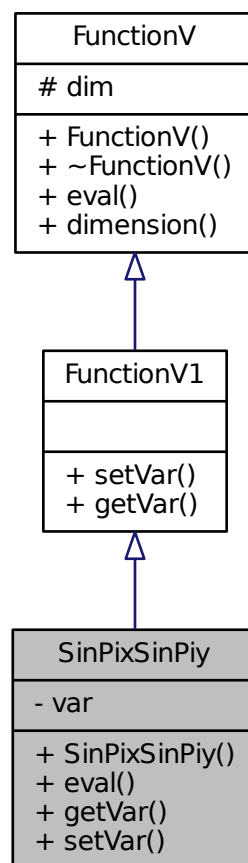
The documentation for this class was generated from the following file:

- [SinPixCosPiy.hpp](#)

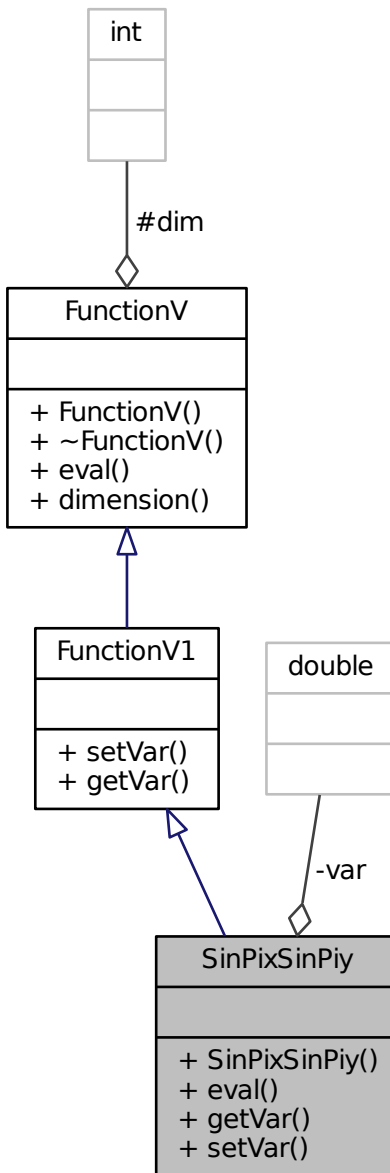
## 7.67 SinPixSinPiy Class Reference

```
#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:

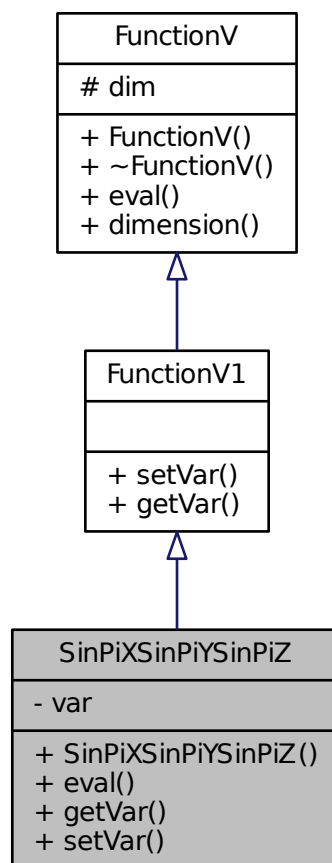
- [SinPixSinPiy.hpp](#)

## 7.68 SinPiXSinPiYSinPiZ Class Reference

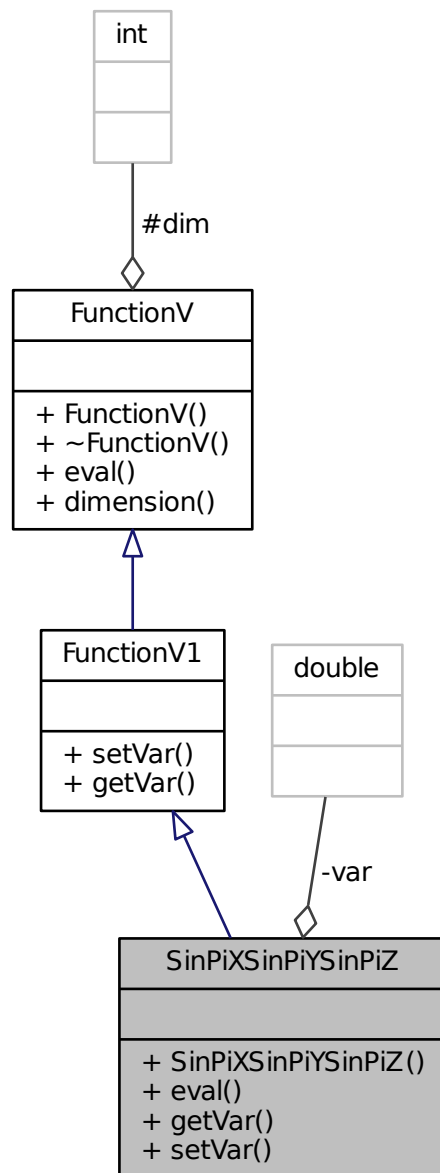
```
#include <SinPiXSinPiYSinPiZ.hpp>
```



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]`

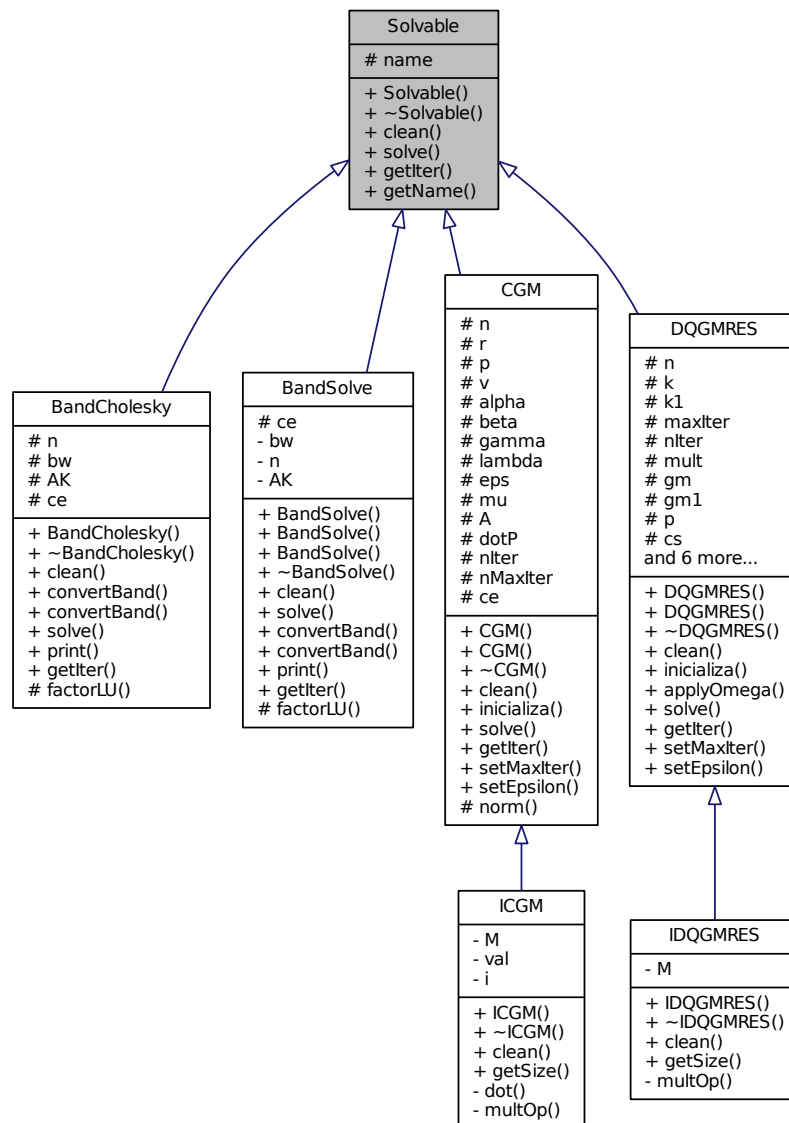
The documentation for this class was generated from the following file:

- [SinPiXSinPiYSinPiZ.hpp](#)

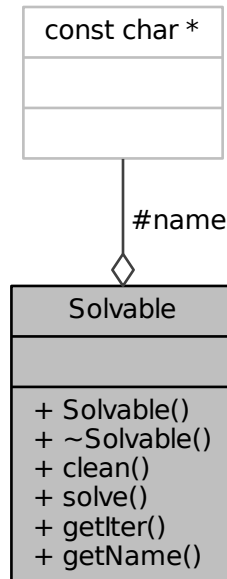
## 7.69 Solvable Class Reference

```
#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]`

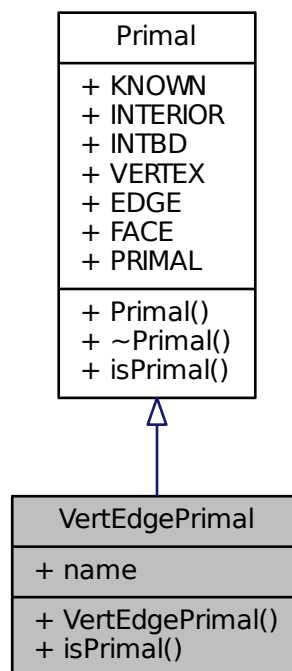
The documentation for this class was generated from the following file:

- [Solvable.hpp](#)

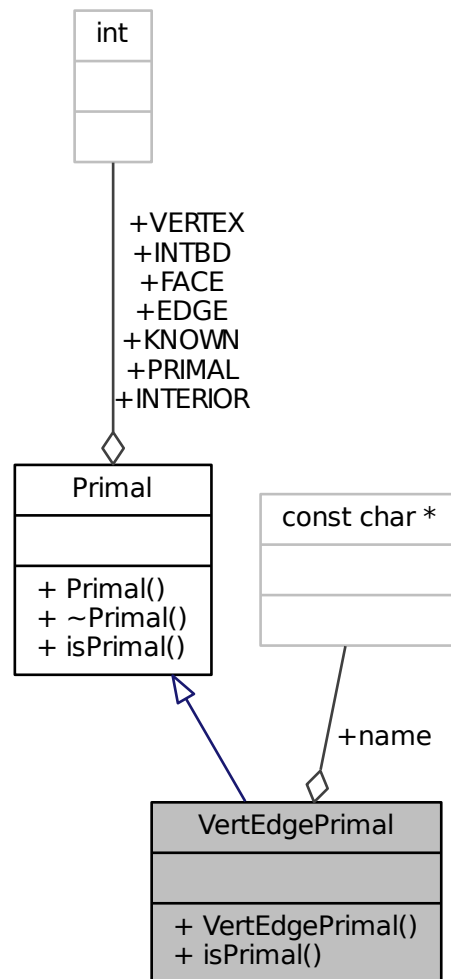
## 7.70 VertEdgePrimal Class Reference

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

Inheritance diagram for VertEdgePrimal:



Collaboration diagram for VertEdgePrimal:



### Public Member Functions

- [VertEdgePrimal](#) (void)
- bool [isPrimal](#) (int type, int \*coordN, int \*coordM)

### Public Attributes

- const char \* [name](#)

### Additional Inherited Members



## 7.70.1 Constructor & Destructor Documentation

7.70.1.1 `VertEdgePrimal::VertEdgePrimal ( void )` `[inline]`

## 7.70.2 Member Function Documentation

7.70.2.1 `bool VertEdgePrimal::isPrimal ( int type, int * coordN, int * coordM )` `[inline]`, `[virtual]`

Implements [Primal](#).

## 7.70.3 Member Data Documentation

7.70.3.1 `const char* VertEdgePrimal::name`

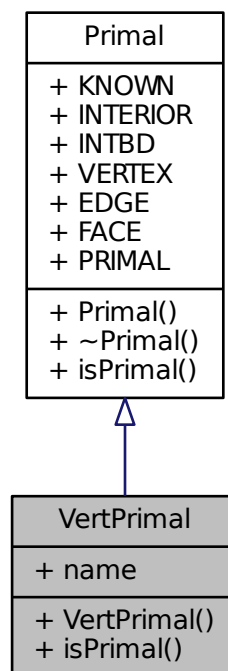
The documentation for this class was generated from the following file:

- [VertEdgePrimal.hpp](#)

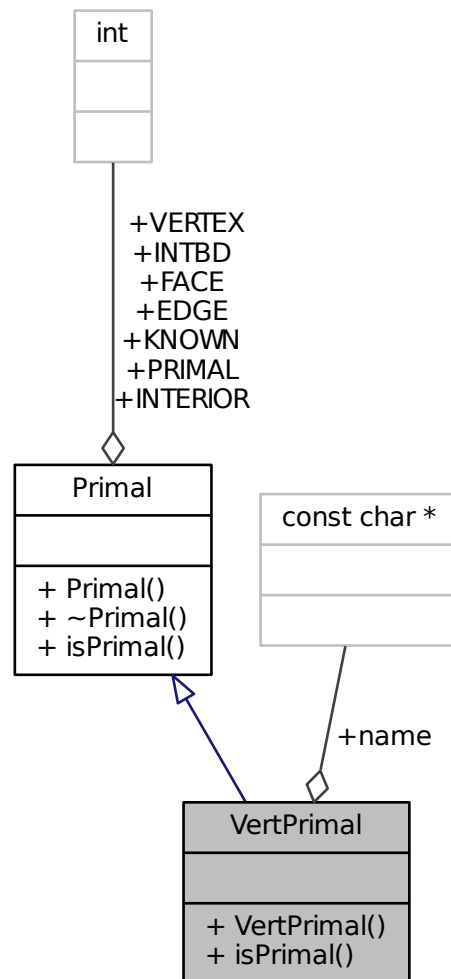
## 7.71 VertPrimal Class Reference

```
#include <VertPrimal.hpp>
```

Inheritance diagram for VertPrimal:



Collaboration diagram for VertPrimal:



### Public Member Functions

- [VertPrimal](#) (void)
- bool [isPrimal](#) (int type, int \*coordN, int \*coordM)

### Public Attributes

- const char \* [name](#)

### Additional Inherited Members

### 7.71.1 Constructor & Destructor Documentation

7.71.1.1 `VertPrimal::VertPrimal ( void )` `[inline]`

### 7.71.2 Member Function Documentation

7.71.2.1 `bool VertPrimal::isPrimal ( int type, int * coordN, int * coordM )` `[inline],[virtual]`

Implements [Primal](#).

### 7.71.3 Member Data Documentation

7.71.3.1 `const char* VertPrimal::name`

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

- [VertPrimal.hpp](#)

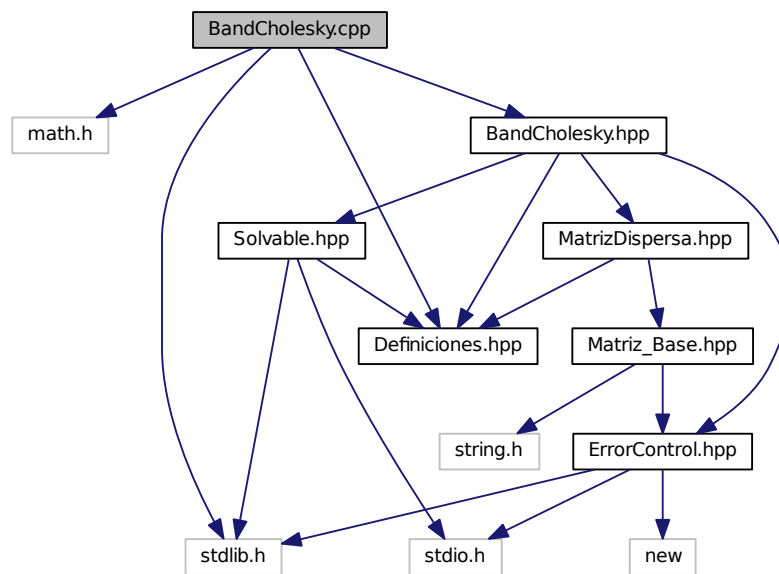
## File Documentation

```
#include "Primal.hpp"
#include <string>
Include dependency graph for AllPrimal.hpp:
```



## 8.2 BandCholesky.cpp File Reference

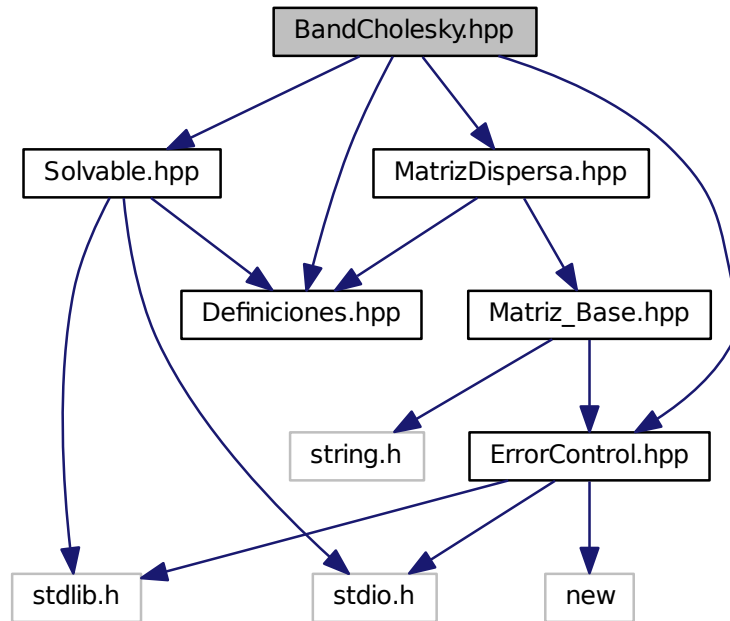
```
#include <math.h>
#include <stdlib.h>
#include "Definiciones.hpp"
#include "BandCholesky.hpp"
Include dependency graph for BandCholesky.cpp:
```



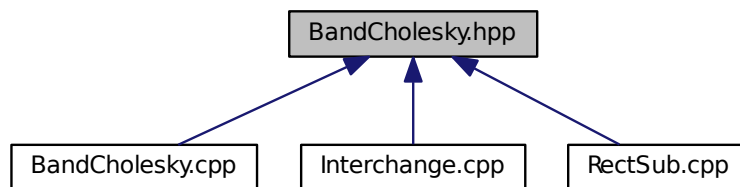
## 8.3 BandCholesky.hpp File Reference

```
#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:

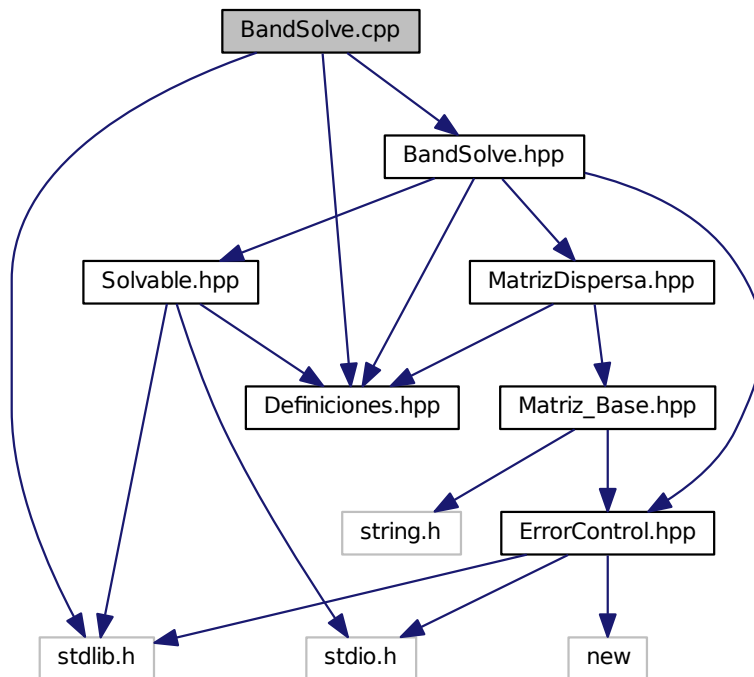


## Classes

- class [BandCholesky](#)

## 8.4 BandSolve.cpp File Reference

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

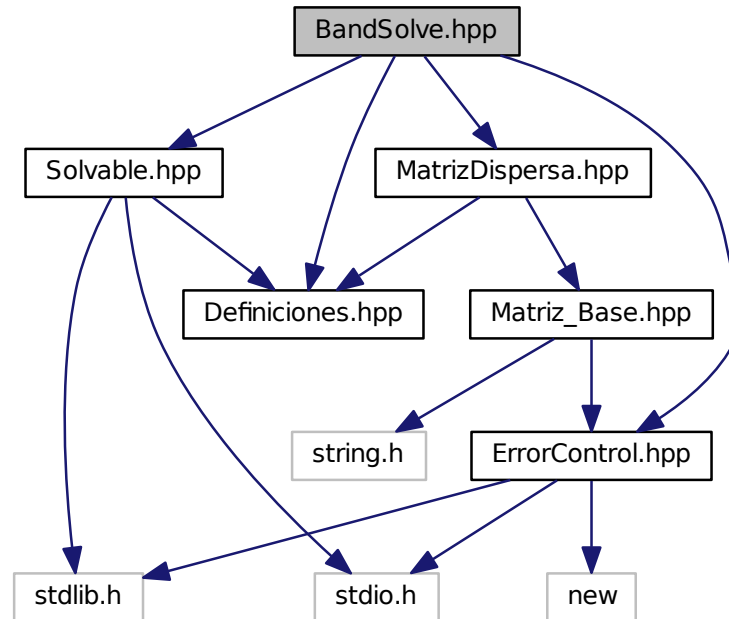


## 8.5 BandSolve.hpp File Reference

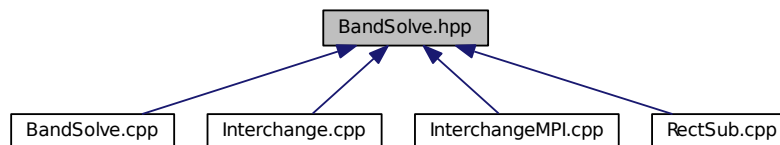
```
#include "Definiciones.hpp"
#include "Solvable.hpp"
#include "MatrizDispersa.hpp"
#include "ErrorControl.hpp"
```



Include dependency graph for BandSolve.hpp:



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

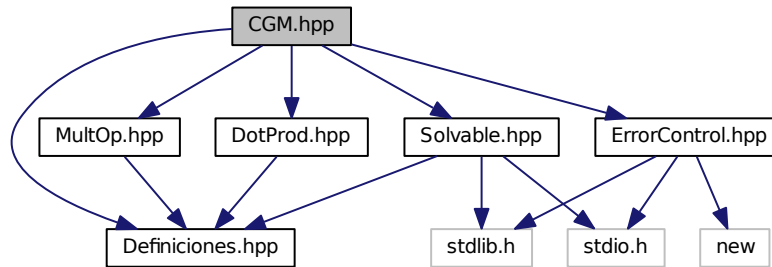


## Classes

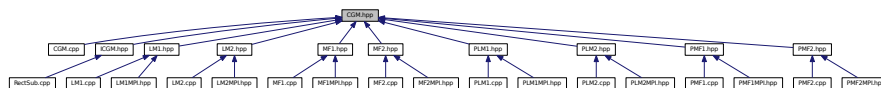
- class [BandSolve](#)



Include dependency graph for CGM.hpp:



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



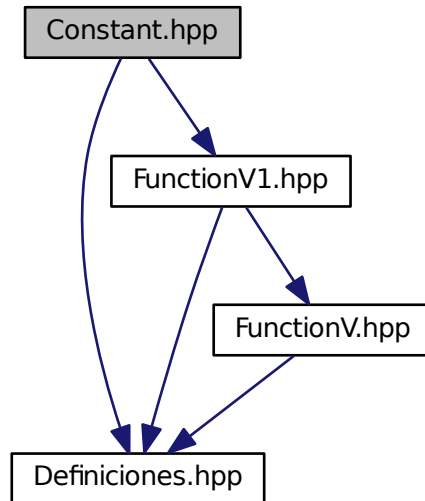
## Classes

- class [CGM](#)

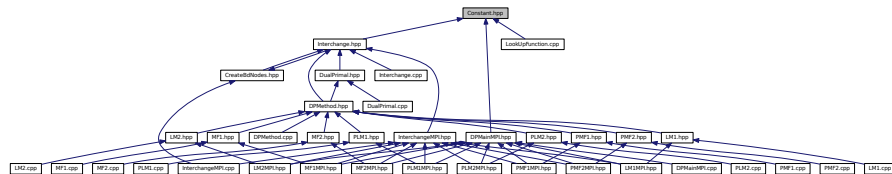
## 8.9 Constant.hpp File Reference

```
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for Constant.hpp:



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



## Classes

- class [Constant](#)

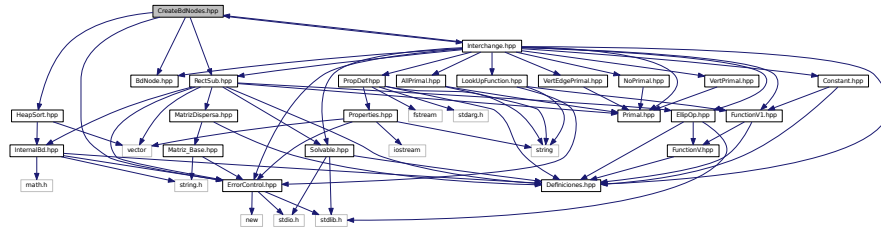
## 8.10 CreateBdNodes.hpp File Reference

```

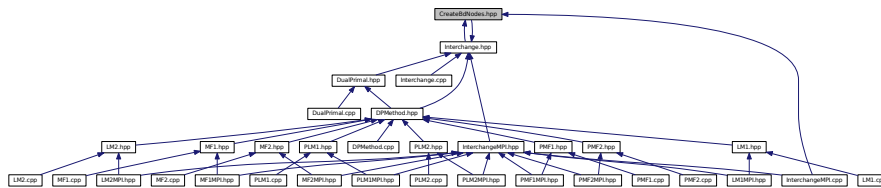
#include "RectSub.hpp"
#include "BdNode.hpp"
#include "HeapSort.hpp"
#include "Interchange.hpp"
#include "ErrorControl.hpp"

```

Include dependency graph for CreateBdNodes.hpp:



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



## Classes

- class [CreateBdNodes](#)

## 8.11 Definiciones.hpp File Reference

### Macros

- #define [NMAXITER](#) 5000  
*Usa la libreria libgmm++-dev.*
- #define [NMAXITER\\_LOCAL](#) 50000
- #define [EPSILON](#) 1e-6  
*Tolerancia en los metodos iterativos.*
- #define [EPSILON\\_LOCAL](#) (EPSILON/1e+4)
- #define [EPS\\_EQUAL](#) 1e-15  
*Se toman como iguales dos nodos que difieran en menos que esta EPS\_EQUAL.*
- #define [DIM\\_VECTOR](#) 1  
*Con esta opcion visualiza o no el residual de cada iteracion.*
- #define [COEFICIENTES\\_CONSTANTES](#)  
*Con esta opcion se calcula el numero de condicionamiento en los metodos 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

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

Dimension del vector (1) escalar

#### 8.11.1.4 #define EPS\_EQUAL 1e-15

Se toman como iguales dos nodos que difieran en menos que esta EPS\_EQUAL.

#### 8.11.1.5 #define EPSILON 1e-6

Tolerancia en los metodos iterativos.

#### 8.11.1.6 #define EPSILON\_LOCAL (EPSILON/1e+4)

#### 8.11.1.7 #define NMAXITER 5000

Usa la libreria libgmm++-dev.

Numero maximo de iteraiones en los metodos iterativos

#### 8.11.1.8 #define NMAXITER\_LOCAL 50000

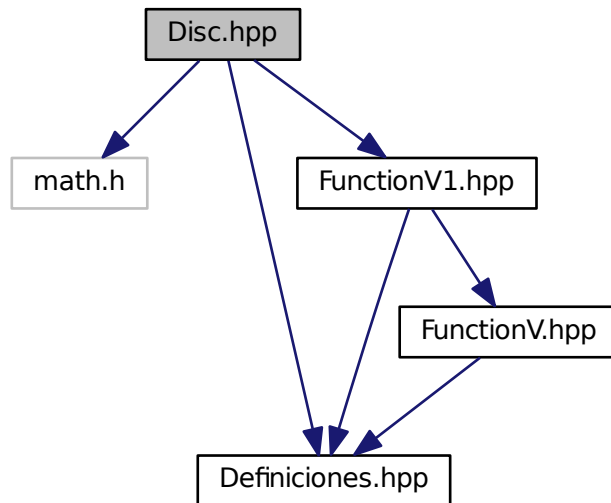
### 8.11.2 Typedef Documentation

#### 8.11.2.1 typedef double ldouble

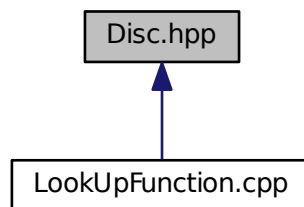
Define ldouble como double.

## 8.12 Disc.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for Disc.hpp:
```



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

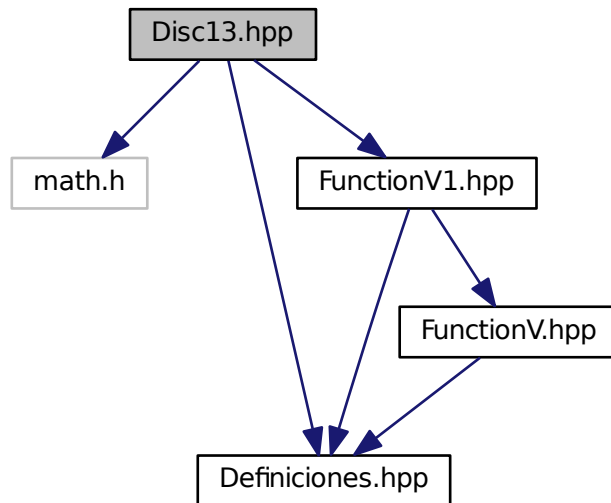


### Classes

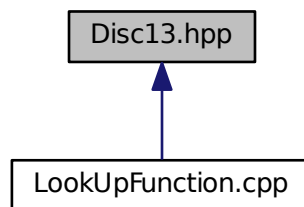
- class [Disc](#)

## 8.13 Disc13.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for Disc13.hpp:
```



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



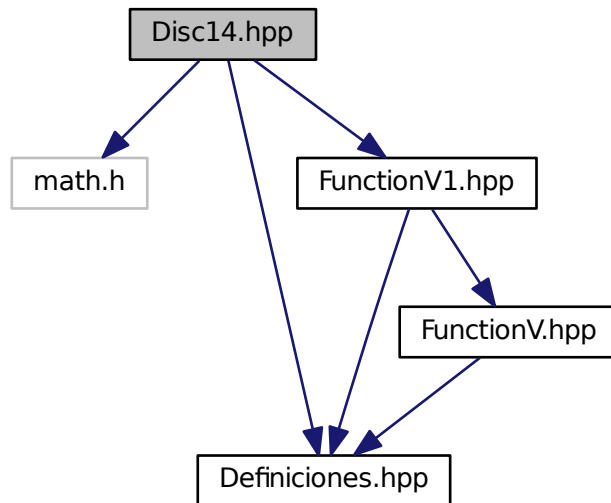
### Classes

- class [Disc13](#)

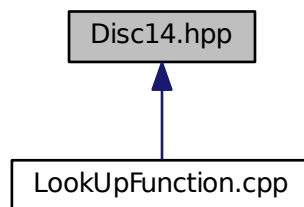


## 8.14 Disc14.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for Disc14.hpp:
```



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

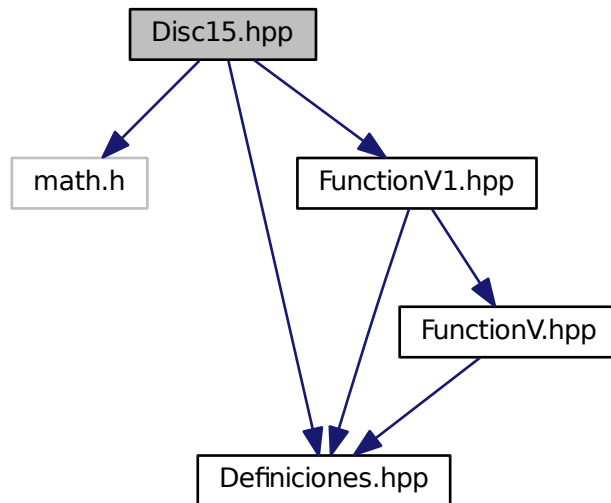


### Classes

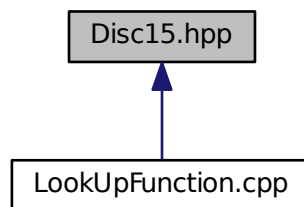
- class [Disc14](#)

## 8.15 Disc15.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for Disc15.hpp:
```



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



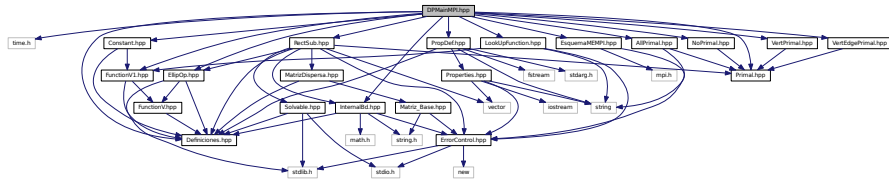
### Classes

- class [Disc15](#)

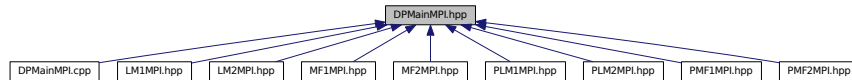


## 8.18 DPMMainMPI.hpp File Reference

```
#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 DPMMainMPI.hpp:
```



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



## Classes

- class [DPMMainMPI](#)

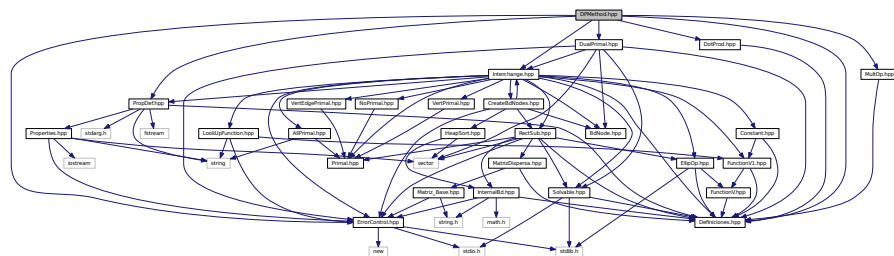
*Clase base para definir a los metodos DVS-DDM.*

## 8.19 DPMMethod.cpp File Reference

```
#include <string>
#include <time.h>
#include "DPMMethod.hpp"
```

[illegible]

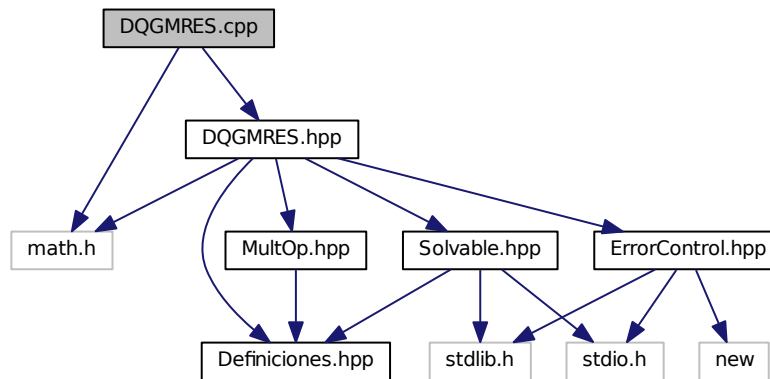
```
#include "Definiciones.hpp"
#include "MultOp.hpp"
#include "DotProd.hpp"
#include "PropDef.hpp"
#include "DualPrimal.hpp"
#include "Interchange.hpp"
#include "ErrorControl.hpp"
Include dependency graph for DPMMethod.hpp:
```



- class **DPMMethod**

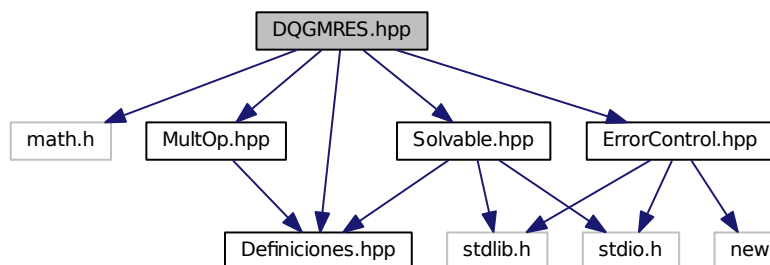
## 8.21 DQGMRES.cpp File Reference

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



## 8.22 DQGMRES.hpp File Reference

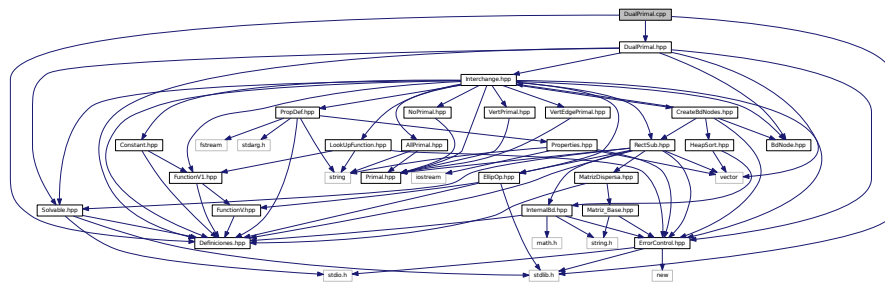
```
#include <math.h>
#include "Definiciones.hpp"
#include "Solvable.hpp"
#include "MultOp.hpp"
#include "ErrorControl.hpp"
Include dependency graph for DQGMRES.hpp:
```



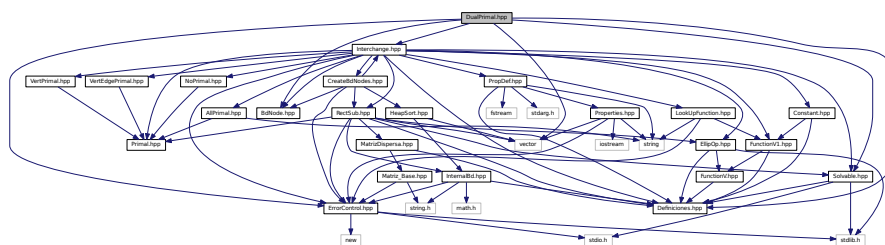
[illegible]

- class DQGMRES

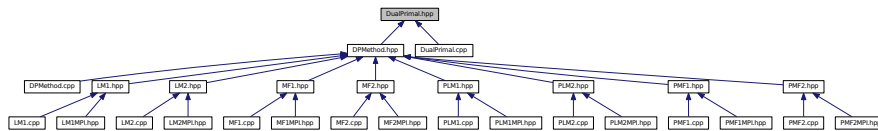
```
#include <stdlib.h>
#include "Definiciones.hpp"
#include "DualPrimal.hpp"
Include dependency graph for DualPrimal.cpp:
```



```
#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:

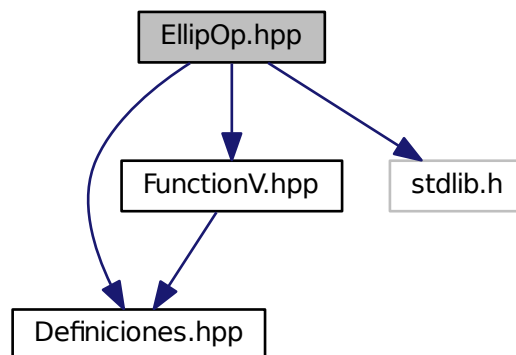


## Classes

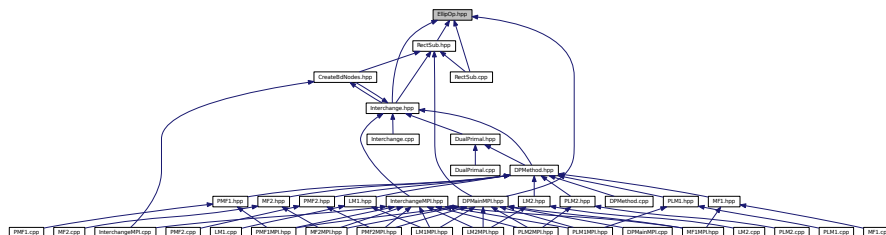
- class DualPrimal

## 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:





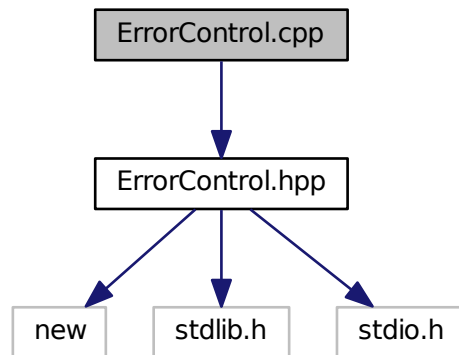
## Classes

- class [EllipOp](#)

## 8.26 ErrorControl.cpp File Reference

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

Include dependency graph for ErrorControl.cpp:



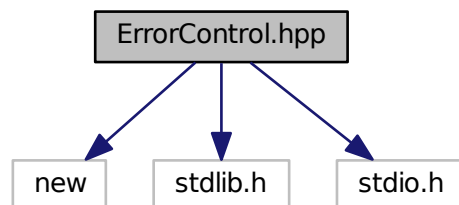
## 8.27 ErrorControl.hpp File Reference

```
#include <new>
```

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

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

Include dependency graph for ErrorControl.hpp:

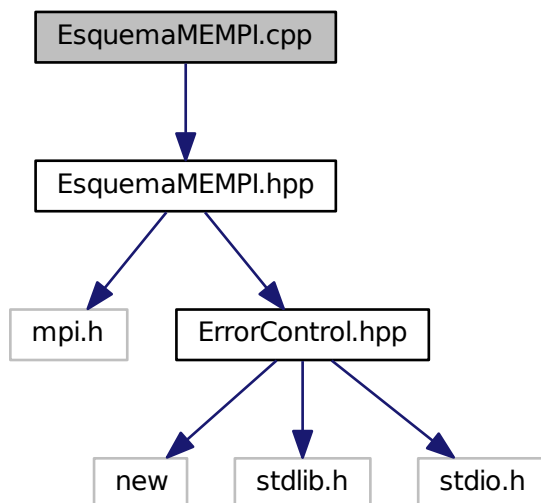


[illegible]

- class **ErrorControl**  
*Error Control.*

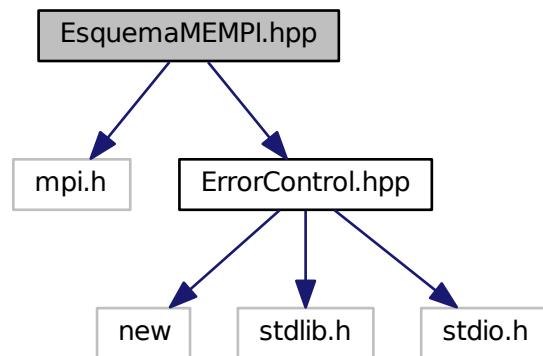
```
#include "EsquemaMEMPI.hpp"
```

Include dependency graph for EsquemaMEMPI.cpp:

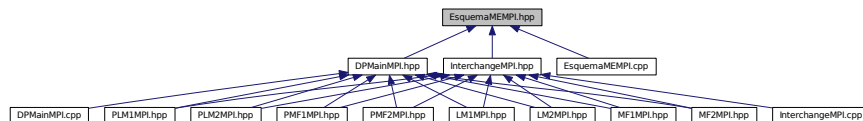


## 8.29 EsquemaMEMPI.hpp File Reference

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



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



### Classes

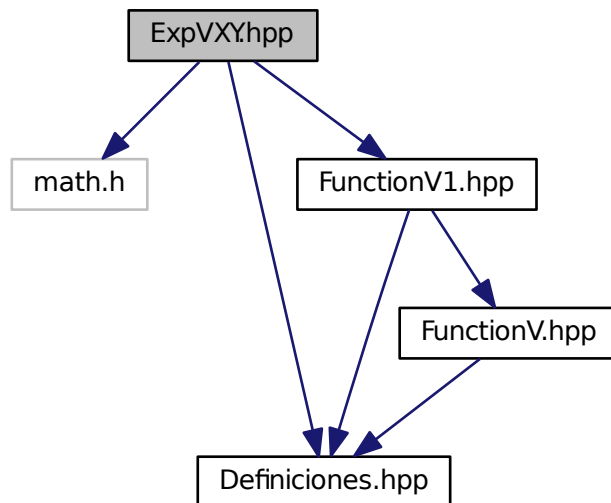
- class [EsquemaMEMPI](#)

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

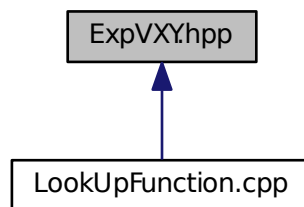
## 8.30 ExpVXY.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for ExpVXY.hpp:



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



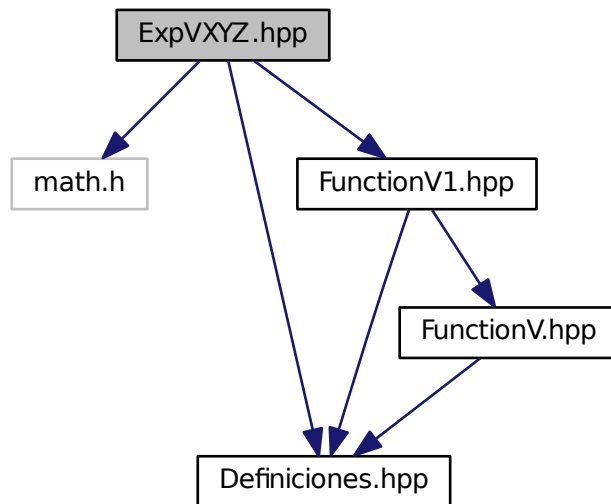
## Classes

- class [ExpVXY](#)

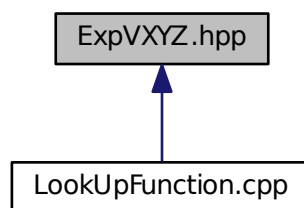
## 8.31 ExpVXYZ.hpp File Reference

```
#include <math.h>
```

```
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for ExpVXYZ.hpp:
```



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

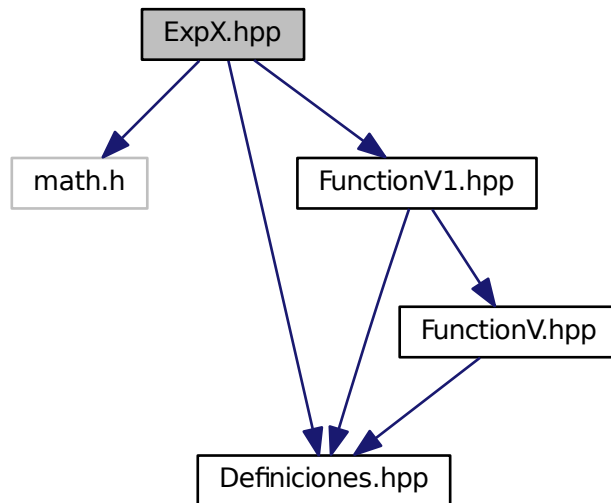


## Classes

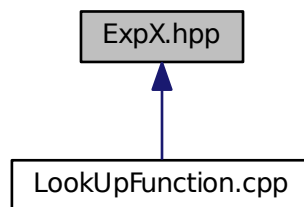
- class [ExpVXYZ](#)

## 8.32 ExpX.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for ExpX.hpp:
```



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

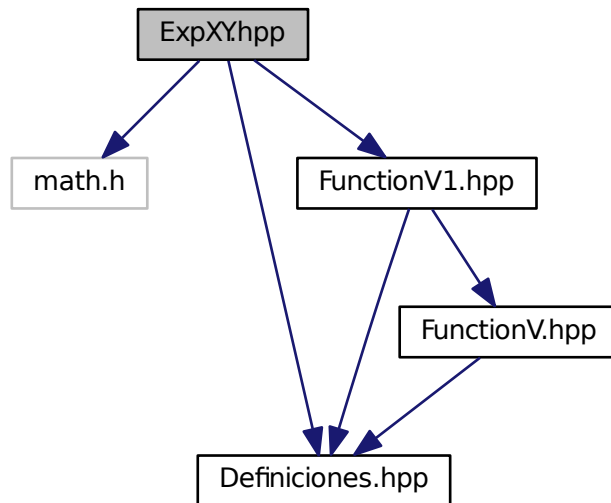


## Classes

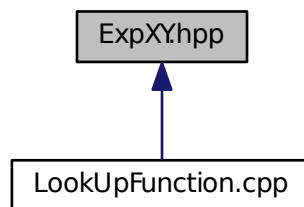
- class [ExpX](#)

## 8.33 ExpXY.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for ExpXY.hpp:
```



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

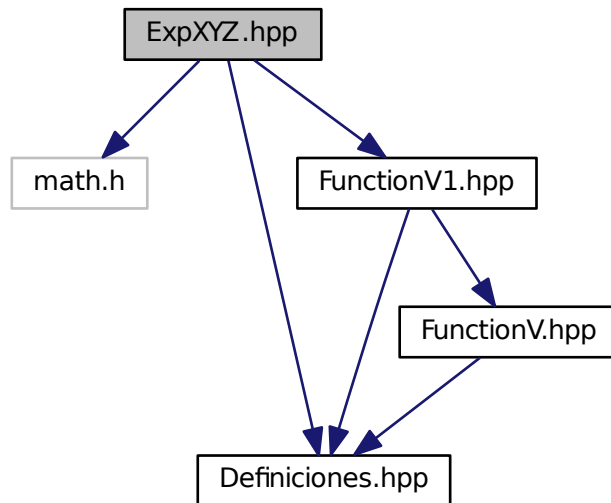


### Classes

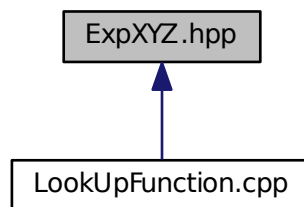
- class [ExpXY](#)

## 8.34 ExpXYZ.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for ExpXYZ.hpp:
```



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



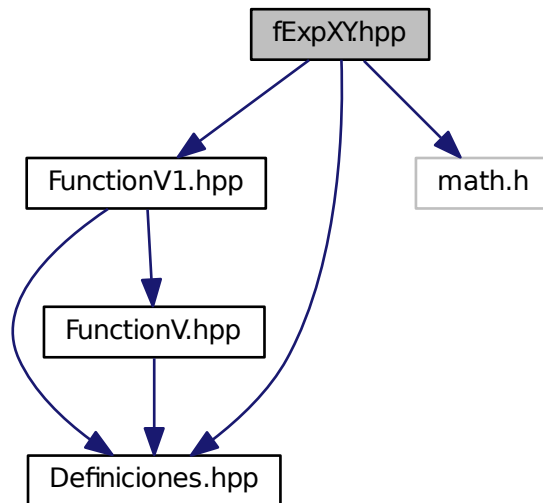
### Classes

- class [ExpXYZ](#)

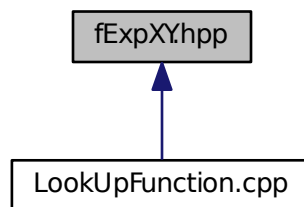


## 8.35 fExpXY.hpp File Reference

```
#include "FunctionV1.hpp"  
#include "Definiciones.hpp"  
#include <math.h>  
Include dependency graph for fExpXY.hpp:
```



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



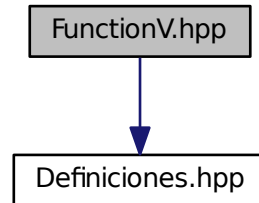
### Classes

- class [fExpXY](#)

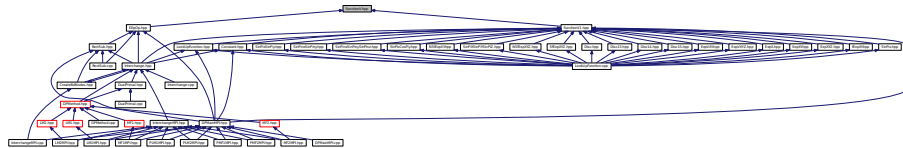
## 8.36 FunctionV.hpp File Reference

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

Include dependency graph for FunctionV.hpp:



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



### Classes

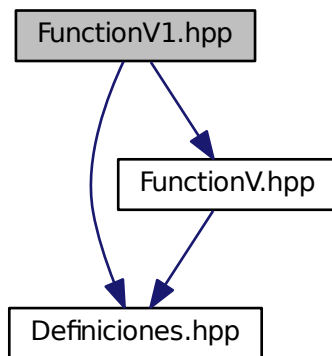
- class [FunctionV](#)

## 8.37 FunctionV1.hpp File Reference

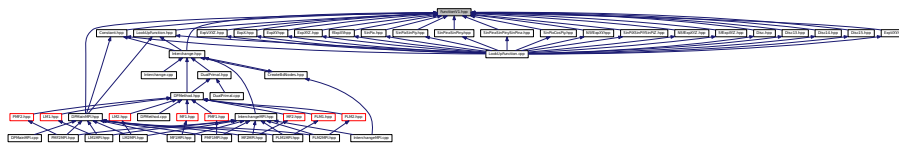
```
#include "Definiciones.hpp"
```

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

Include dependency graph for FunctionV1.hpp:



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



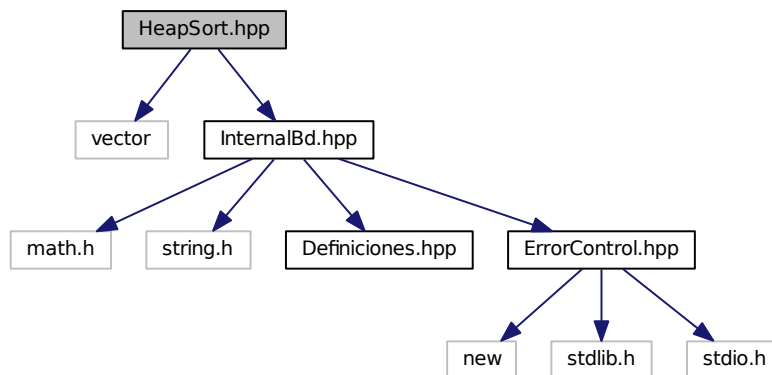
## Classes

- class [FunctionV1](#)

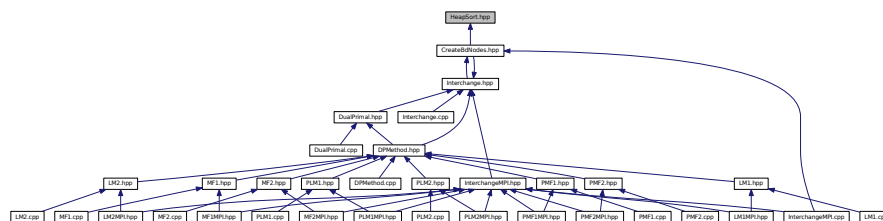
## 8.38 HeapSort.hpp File Reference

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

Include dependency graph for HeapSort.hpp:



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



## Classes

- class [HeapSort](#)

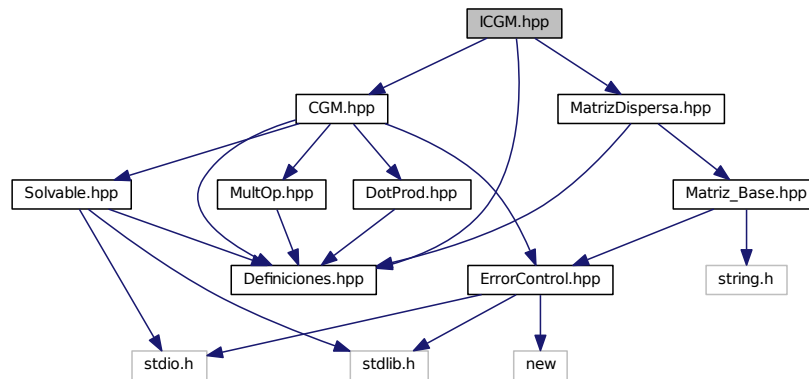
## 8.39 ICGM.hpp File Reference

```

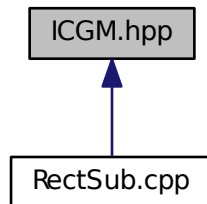
#include "Definiciones.hpp"
#include "CGM.hpp"
#include "MatrizDispersa.hpp"

```

Include dependency graph for ICGM.hpp:



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



## Classes

- class [ICGM](#)

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

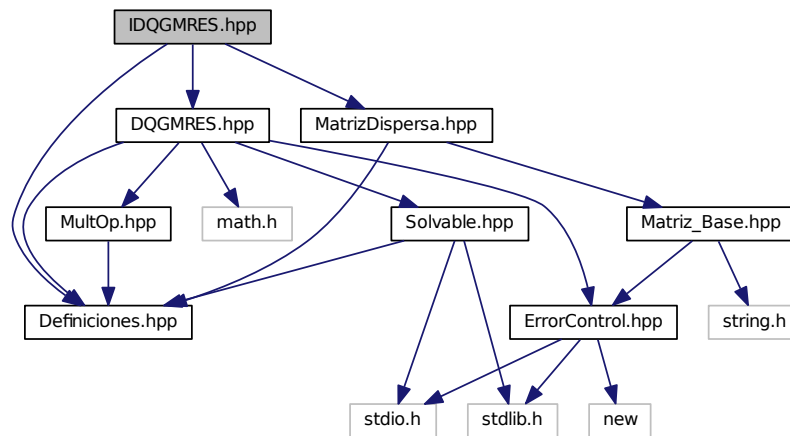
## 8.40 IDQGMRES.hpp File Reference

```

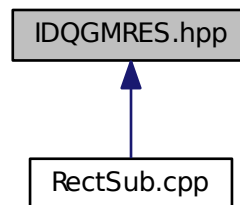
#include "Definiciones.hpp"
#include "DQGMRES.hpp"
#include "MatrizDispersa.hpp"

```

Include dependency graph for IDQGMRES.hpp:



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



## Classes

- class [IDQGMRES](#)

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

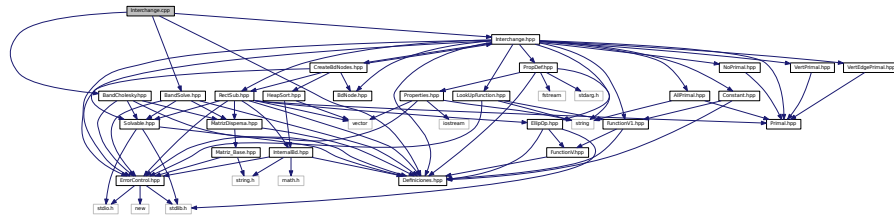
## 8.41 Interchange.cpp File Reference

```

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

```

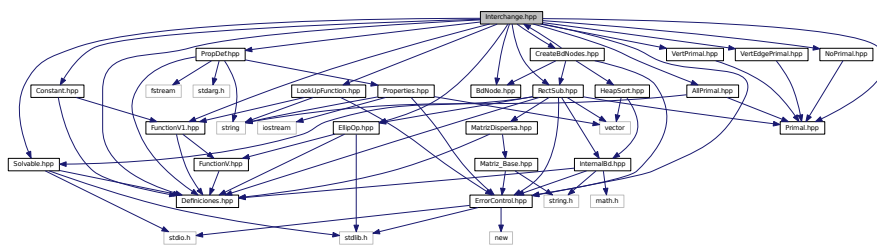
Include dependency graph for Interchange.cpp:



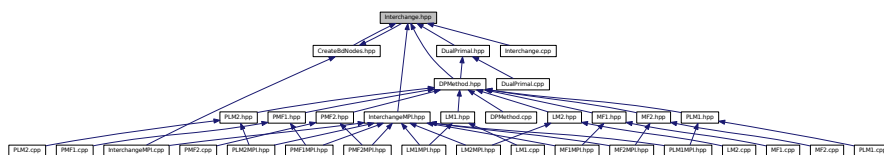
## 8.42 Interchange.hpp File Reference

```
#include "Definiciones.hpp"
#include "BdNode.hpp"
#include "Solvable.hpp"
#include "PropDef.hpp"
#include "RectSub.hpp"
#include "FunctionV1.hpp"
#include "Constant.hpp"
#include "EllipOp.hpp"
#include "Primal.hpp"
#include "LookUpFunction.hpp"
#include "VertPrimal.hpp"
#include "VertEdgePrimal.hpp"
#include "AllPrimal.hpp"
#include "NoPrimal.hpp"
#include "CreateBdNodes.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for Interchange.hpp:



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

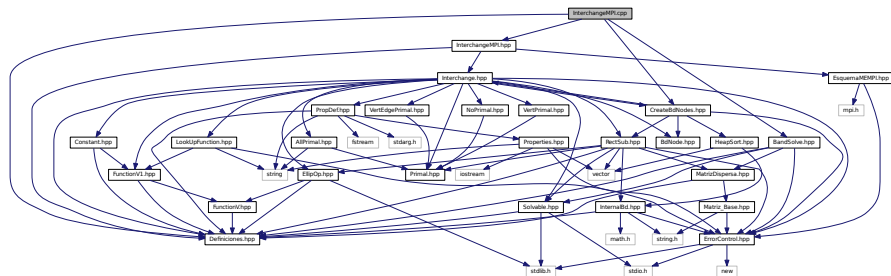


## Classes

- class [Interchange](#)

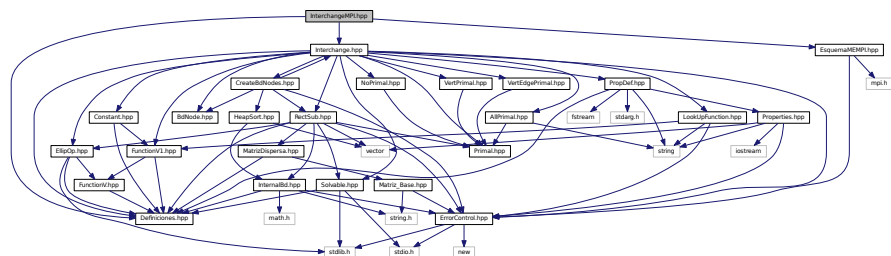
## 8.43 InterchangeMPI.cpp File Reference

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

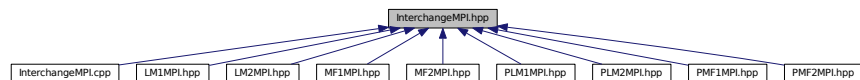


## 8.44 InterchangeMPI.hpp File Reference

```
#include "Definiciones.hpp"
#include "Interchange.hpp"
#include "EsquemaMEMPI.hpp"
Include dependency graph for InterchangeMPI.hpp:
```



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





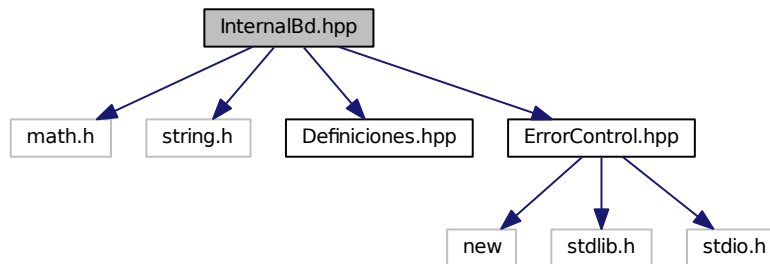
## Classes

- class [InterchangeMPI](#)

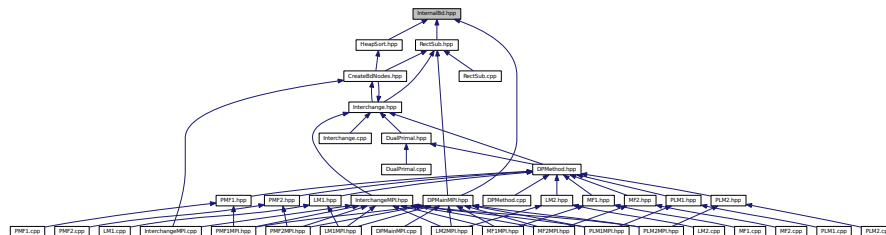
## 8.45 InternalBd.hpp File Reference

```
#include <math.h>
#include <string.h>
#include "Definiciones.hpp"
#include "ErrorControl.hpp"
```

Include dependency graph for InternalBd.hpp:



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



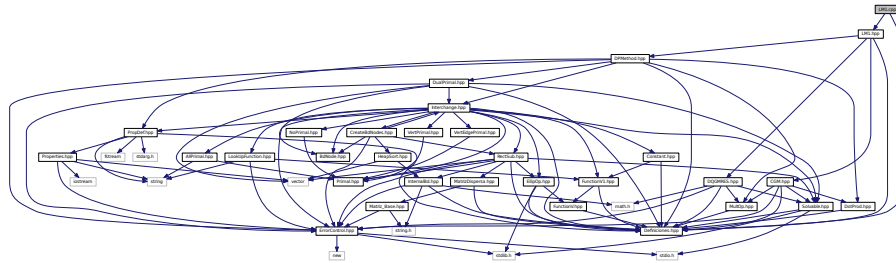
## Classes

- class [InternalBd](#)

## 8.46 LM1.cpp File Reference

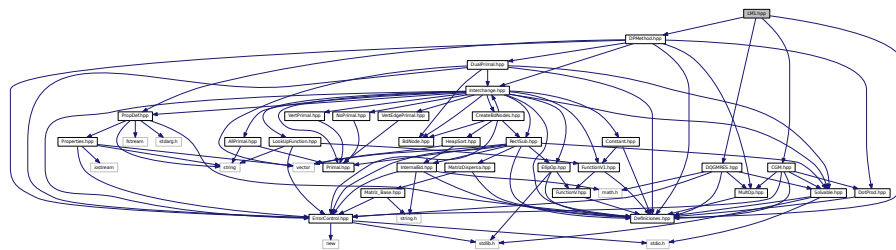
```
#include "Definiciones.hpp"
#include "LM1.hpp"
```

Include dependency graph for LM1.cpp:

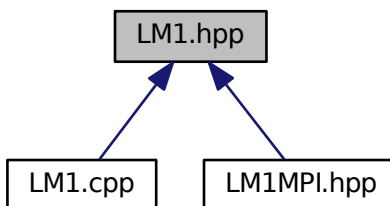


## 8.47 LM1.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
Include dependency graph for LM1.hpp:
```



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

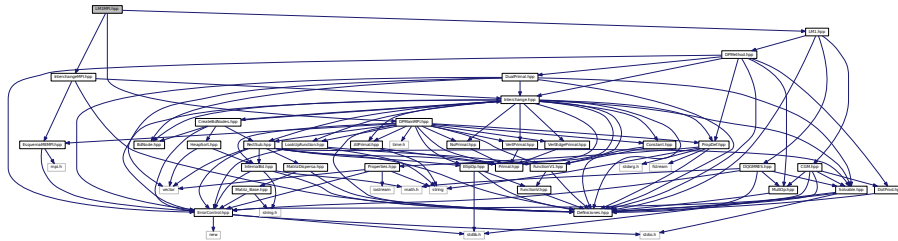


## Classes

- class [LM1](#)

## 8.48 LM1MPI.hpp File Reference

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



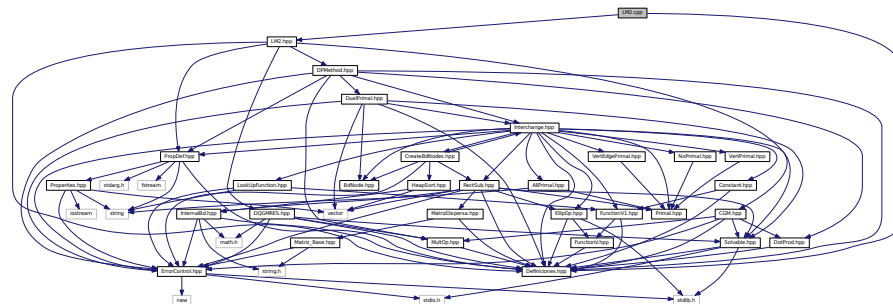
### Classes

- class [LM1MPI](#)

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

## 8.49 LM2.cpp File Reference

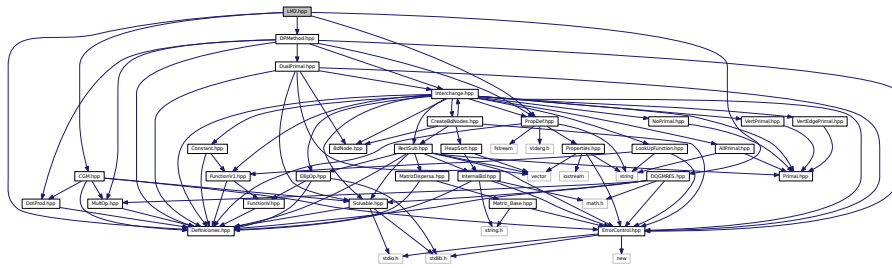
```
#include "Definiciones.hpp"
#include "LM2.hpp"
Include dependency graph for LM2.cpp:
```



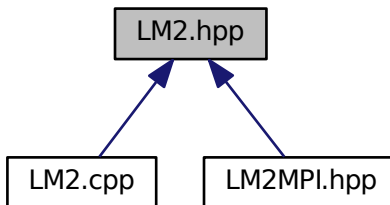
## 8.50 LM2.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "PropDef.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
```

Include dependency graph for LM2.hpp:



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



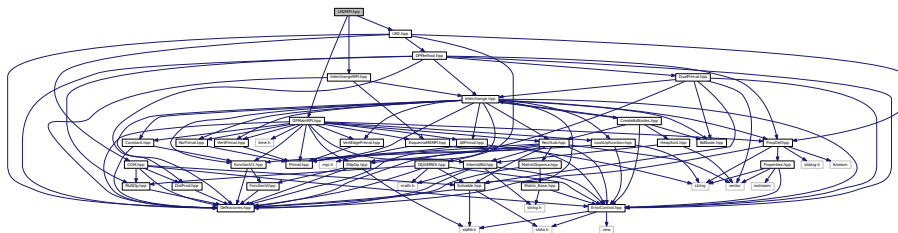
## Classes

- class [LM2](#)

## 8.51 LM2MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "LM2.hpp"
```

Include dependency graph for LM2MPI.hpp:



## Classes

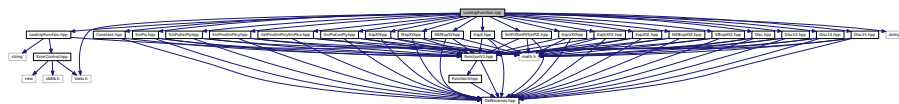
- class [LM2MPI](#)

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

## 8.52 LookUpFunction.cpp File Reference

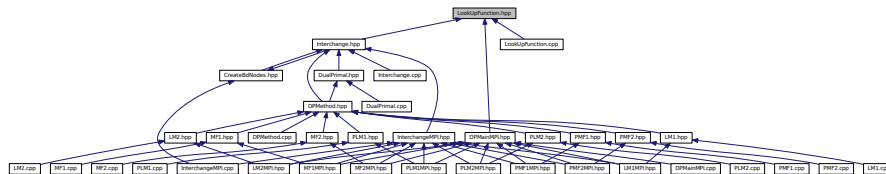
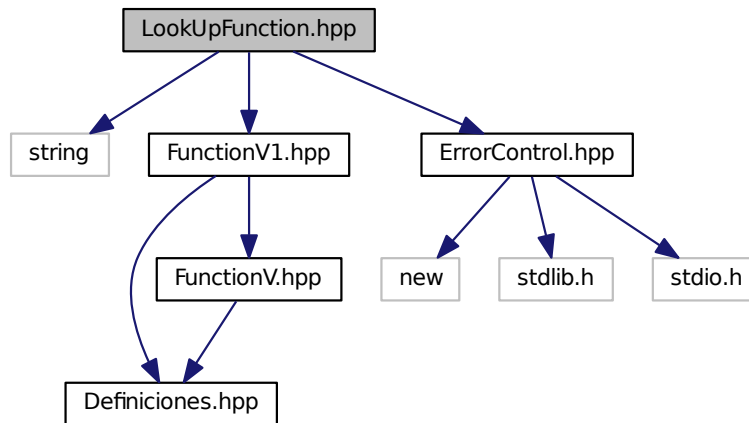
```
#include "LookUpFunction.hpp"
#include "Constant.hpp"
#include "SinPix.hpp"
#include "SinPixSinPiy.hpp"
#include "SinPinxSinPiny.hpp"
#include "SinPinxSinPinySinPinz.hpp"
#include "SinPixCosPiy.hpp"
#include "ExpXY.hpp"
#include "fExpXY.hpp"
#include "NSfExpXY.hpp"
#include "ExpX.hpp"
#include "SinPiXSinPiYSinPiZ.hpp"
#include "ExpVXY.hpp"
#include "ExpVXYZ.hpp"
#include "ExpXYZ.hpp"
#include "NSfExpXYZ.hpp"
#include "SfExpXYZ.hpp"
#include "Disc.hpp"
#include "Disc13.hpp"
#include "Disc14.hpp"
#include "Disc15.hpp"
#include <string.h>
#include <stdio.h>
```

Include dependency graph for LookUpFunction.cpp:



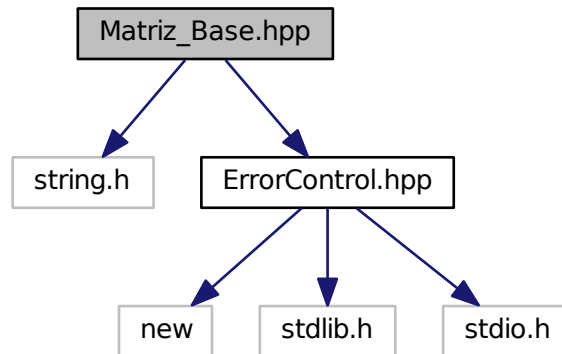
## 8.53 LookUpFunction.hpp File Reference

```
#include <string>
#include "FunctionV1.hpp"
#include "ErrorControl.hpp"
```

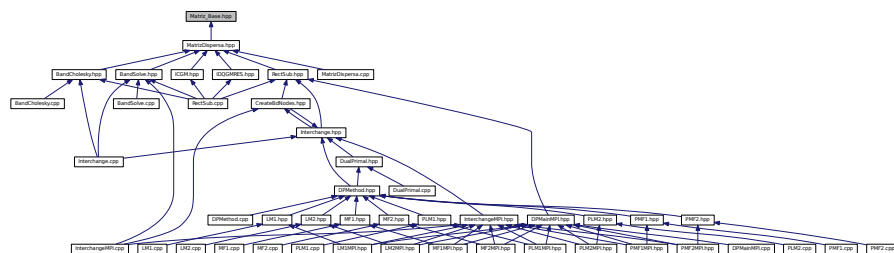


---

Include dependency graph for Matriz\_Base.hpp:



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



## Classes

- class [Matriz\\_Base](#)

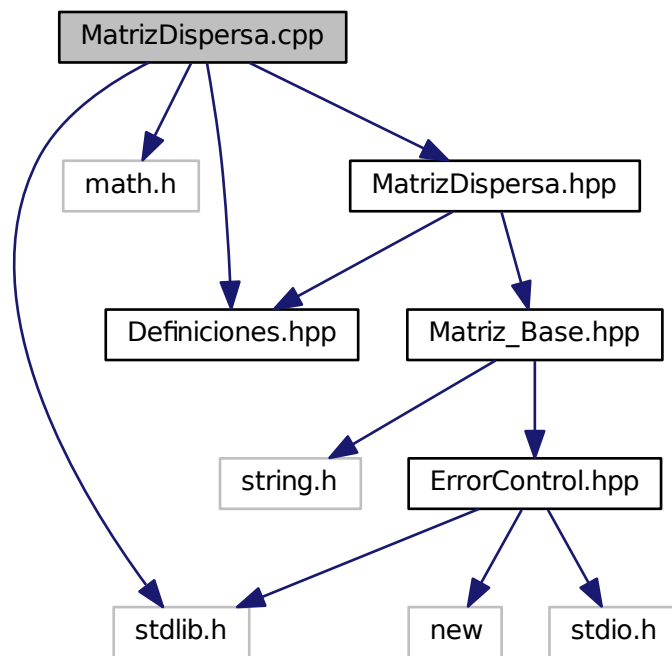
*Clase base para el trabajar con matrices.*

## 8.56 MatrizDispersa.cpp File Reference

```

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

Include dependency graph for MatrizDispersa.cpp:

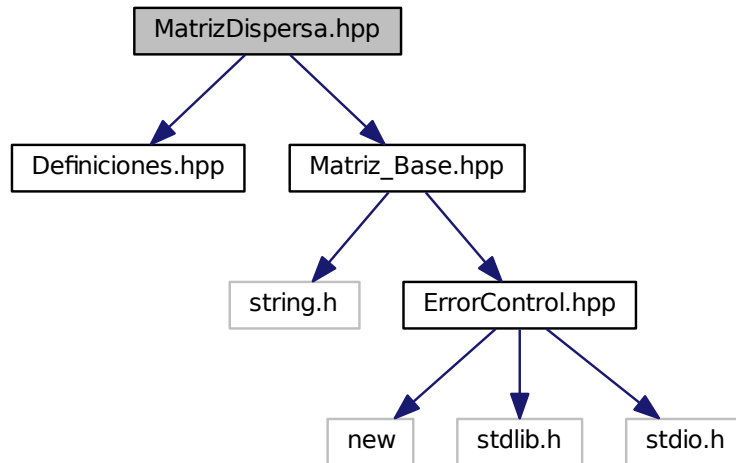


## 8.57 MatrizDispersa.hpp File Reference

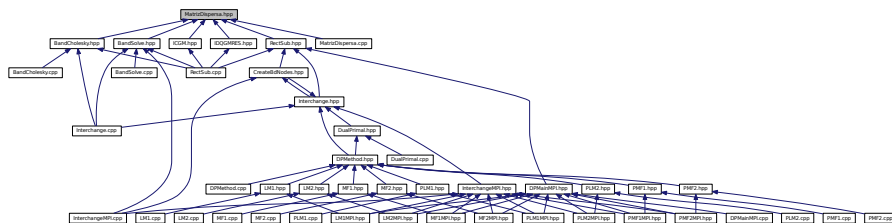
```
#include "Definiciones.hpp"
#include "Matriz_Base.hpp"
```



Include dependency graph for MatrizDispersa.hpp:



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



## Classes

- class [MatrizDispersa](#)

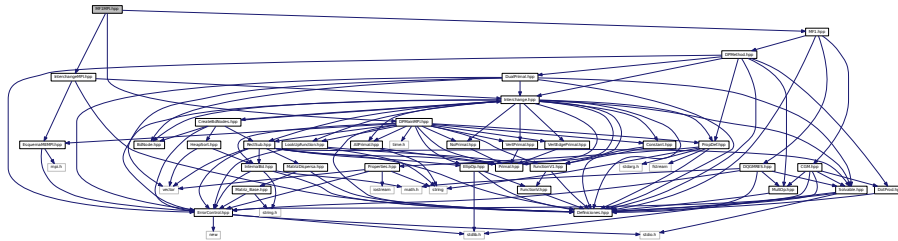
## 8.58 MF1.cpp File Reference

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



## 8.60 MF1MPI.hpp File Reference

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



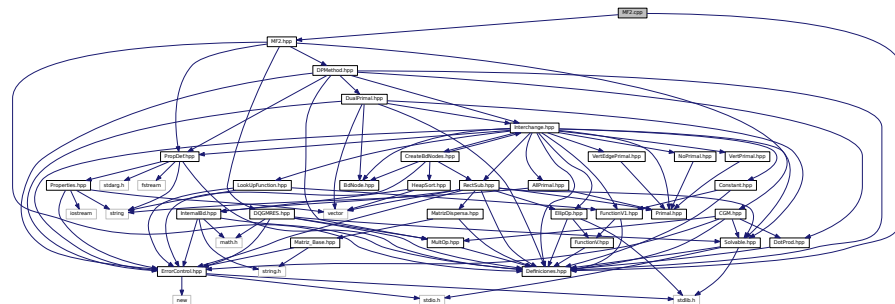
### Classes

- class [MF1MPI](#)

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

## 8.61 MF2.cpp File Reference

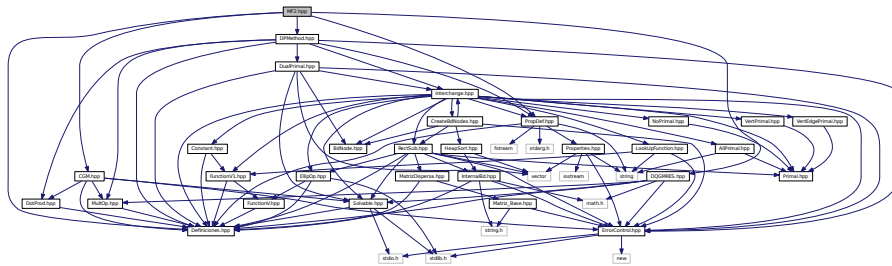
```
#include "Definiciones.hpp"
#include "MF2.hpp"
Include dependency graph for MF2.cpp:
```



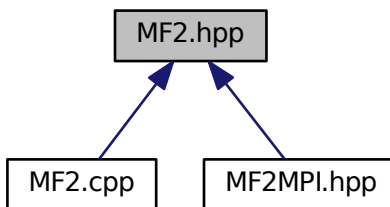
## 8.62 MF2.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMMethod.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:



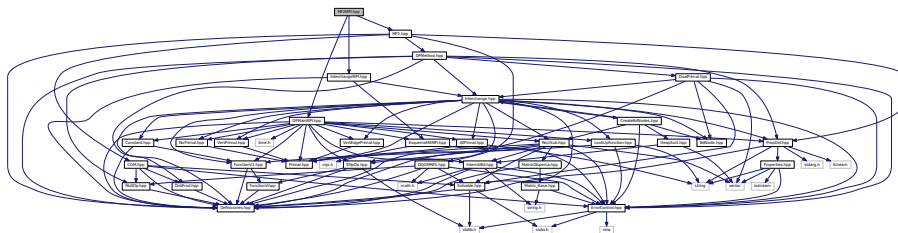
## Classes

- class [MF2](#)

## 8.63 MF2MPI.hpp File Reference

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

Include dependency graph for MF2MPI.hpp:



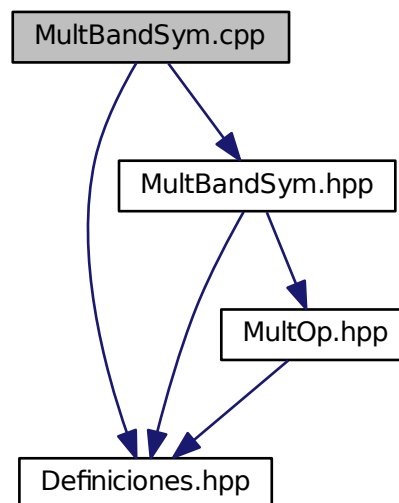
## Classes

- class [MF2MPI](#)

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

## 8.64 MultBandSym.cpp File Reference

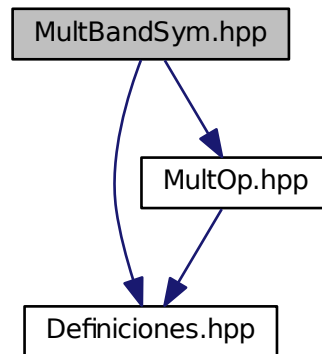
```
#include "Definiciones.hpp"
#include "MultBandSym.hpp"
Include dependency graph for MultBandSym.cpp:
```



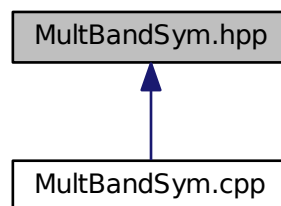
## 8.65 MultBandSym.hpp File Reference

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

Include dependency graph for MultBandSym.hpp:



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



## Classes

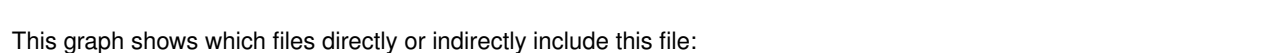
- class [MultBandSym](#)

## 8.66 MultOp.hpp File Reference

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

```
graph TD; A[MultOp.hpp] --> B[Definiciones.hpp];
```

A diagram illustrating a file inclusion relationship. At the top, a box labeled "MultOp.hpp" has a downward-pointing arrow leading to a box labeled "Definiciones.hpp".



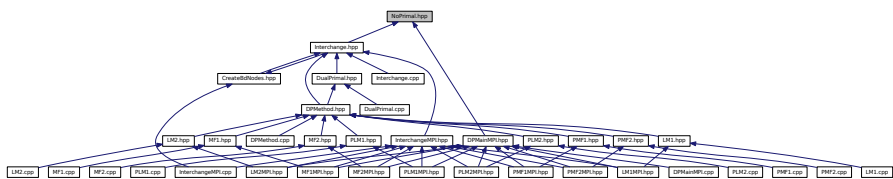
```
#include "Primal.hpp"
```

Include dependency graph for NoPrimal.hpp:

Include dependency graph for NoPrimal.hpp:



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

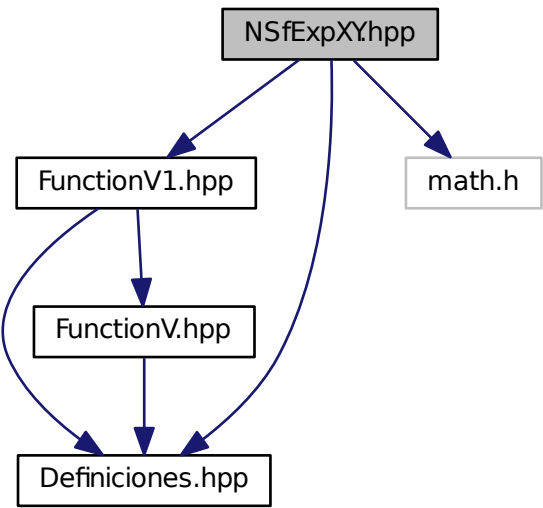


Classes

- class [NoPrimal](#)

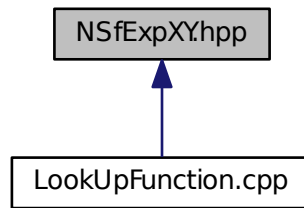
8.68 NSfExpXY.hpp File Reference

```
#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:



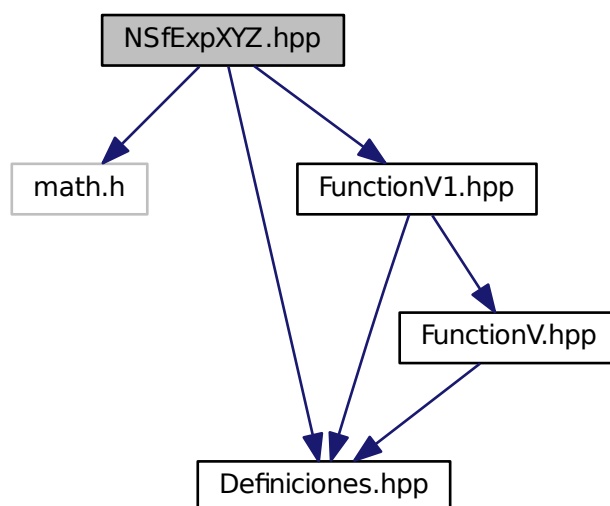
### Classes

- class [NSfExpXY](#)

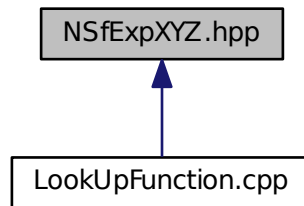
## 8.69 NSfExpXYZ.hpp File Reference

```
#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:

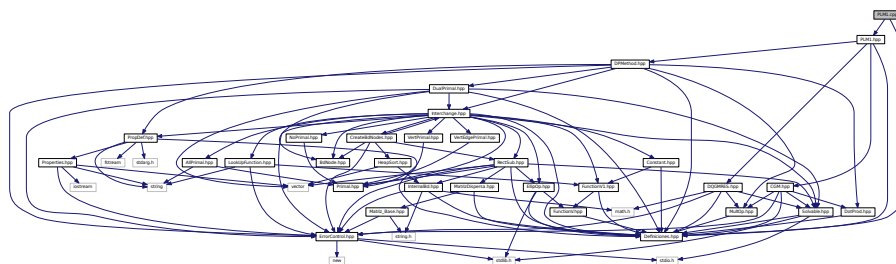


## Classes

- class [NSfExpXYZ](#)

## 8.70 PLM1.cpp File Reference

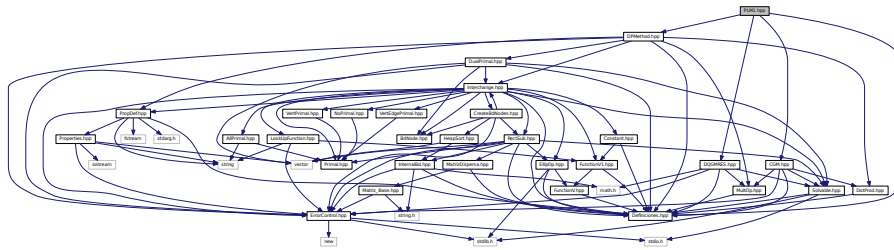
```
#include "Definiciones.hpp"
#include "PLM1.hpp"
Include dependency graph for PLM1.cpp:
```



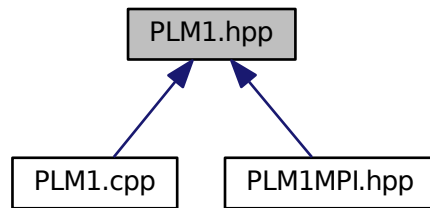
## 8.71 PLM1.hpp File Reference

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

Include dependency graph for PLM1.hpp:



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



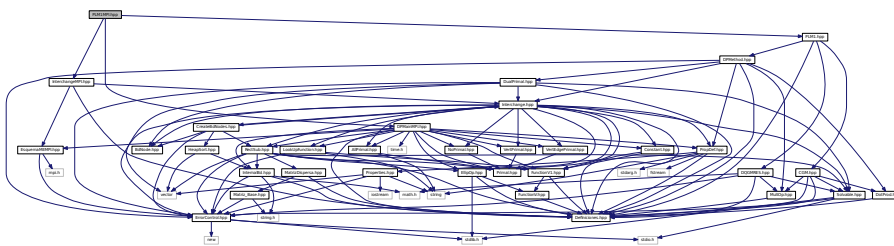
## Classes

- class [PLM1](#)

## 8.72 PLM1MPI.hpp File Reference

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

Include dependency graph for PLM1MPI.hpp:



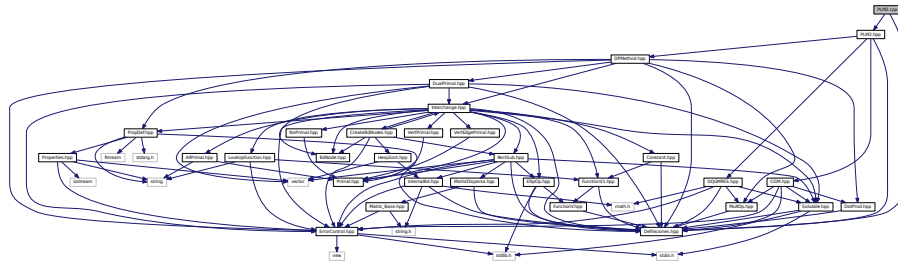
## Classes

- class [PLM1MPI](#)

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

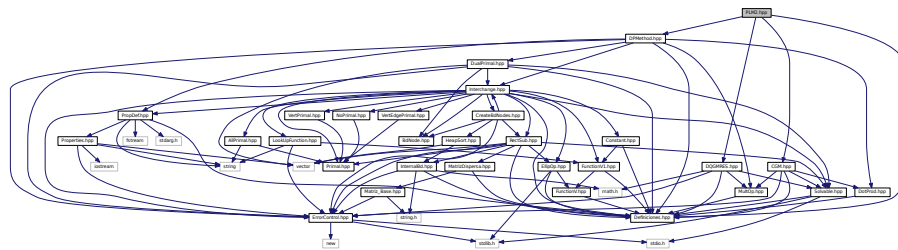
## 8.73 PLM2.cpp File Reference

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

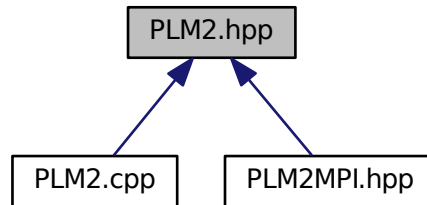


## 8.74 PLM2.hpp File Reference

```
#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:

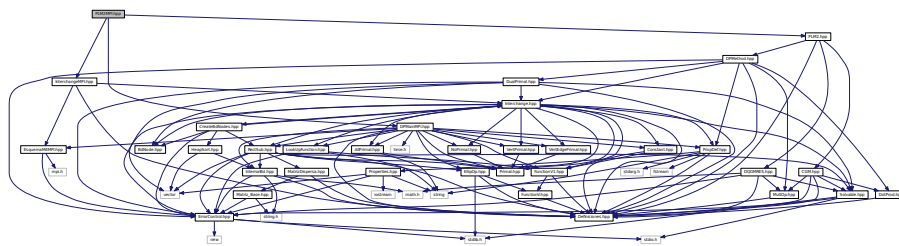


## Classes

- class [PLM2](#)

## 8.75 PLM2MPI.hpp File Reference

```
#include "DPMainMPI.hpp"
#include "InterchangeMPI.hpp"
#include "PLM2.hpp"
Include dependency graph for PLM2MPI.hpp:
```



## Classes

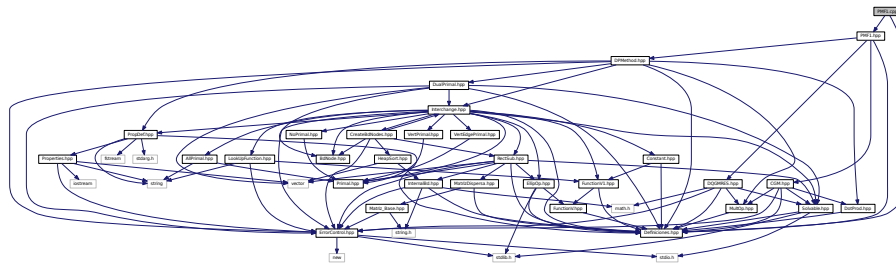
- class [PLM2MPI](#)

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

## 8.76 PMF1.cpp File Reference

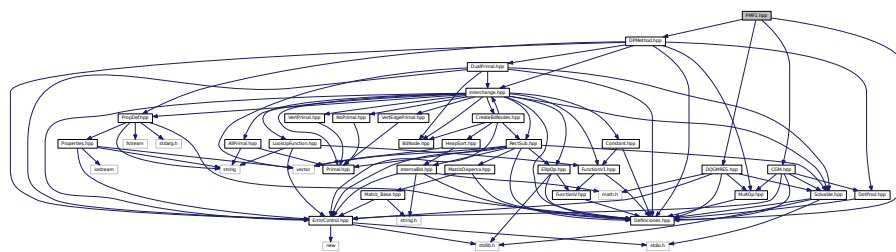
```
#include "Definiciones.hpp"
#include "PMF1.hpp"
```

Include dependency graph for PMF1.cpp:

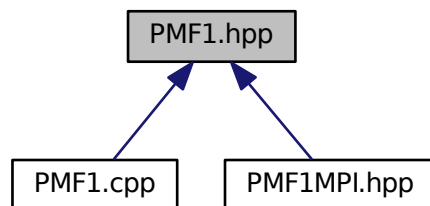


## 8.77 PMF1.hpp File Reference

```
#include "Definiciones.hpp"
#include "DPMethod.hpp"
#include "CGM.hpp"
#include "DQGMRES.hpp"
Include dependency graph for PMF1.hpp:
```



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

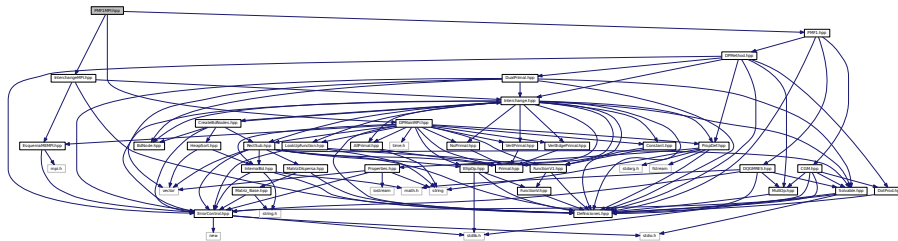


## Classes

- class [PMF1](#)

## 8.78 PMF1MPI.hpp File Reference

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



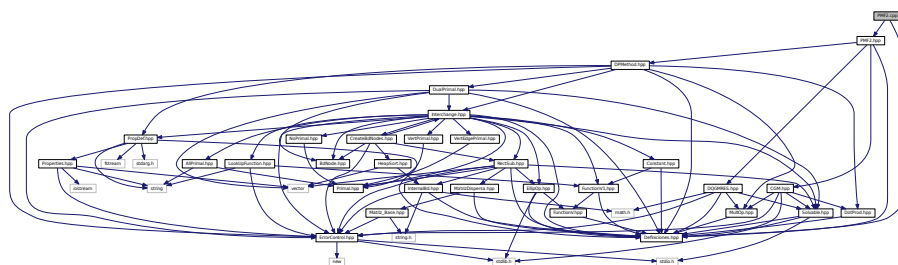
### Classes

- class [PMF1MPI](#)

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

## 8.79 PMF2.cpp File Reference

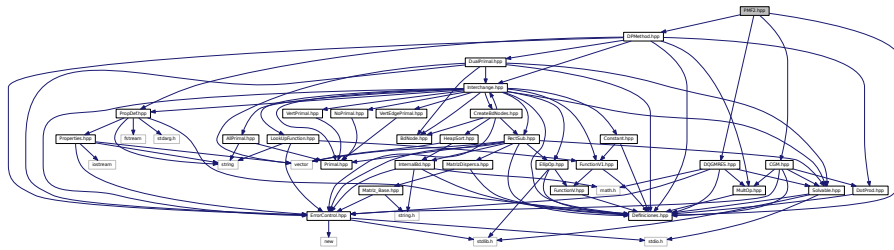
```
#include "Definiciones.hpp"
#include "PMF2.hpp"
Include dependency graph for PMF2.cpp:
```



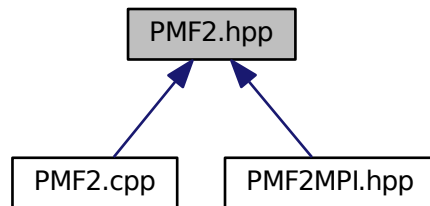
## 8.80 PMF2.hpp File Reference

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

Include dependency graph for PMF2.hpp:



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



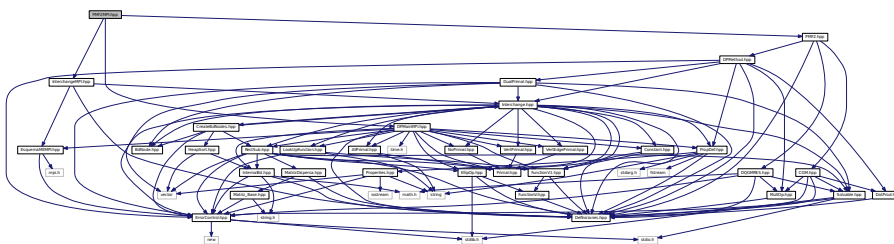
## Classes

- class [PMF2](#)

## 8.81 PMF2MPI.hpp File Reference

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

Include dependency graph for PMF2MPI.hpp:

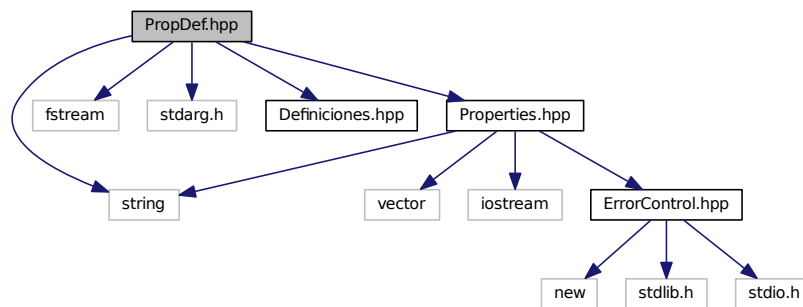




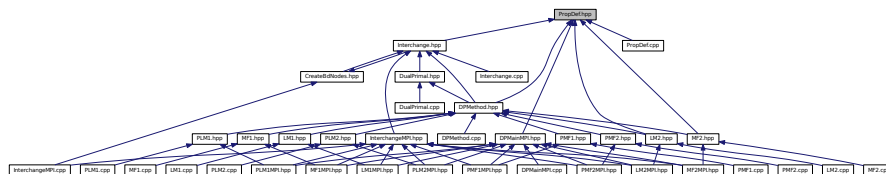


## 8.84 PropDef.hpp File Reference

```
#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:



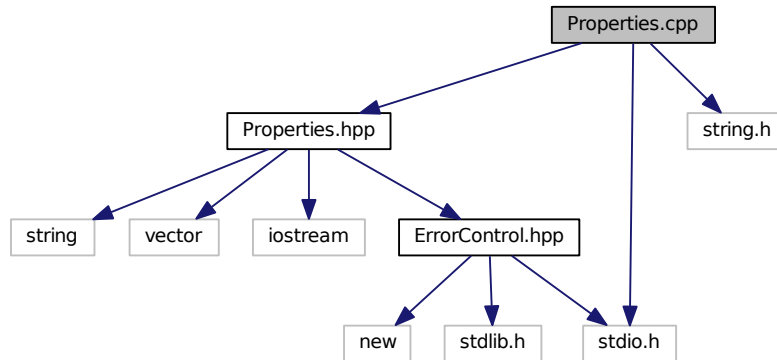
## Classes

- class PropDef

## 8.85 Properties.cpp File Reference

```
#include "Properties.hpp"
#include <string.h>
#include <stdio.h>
```

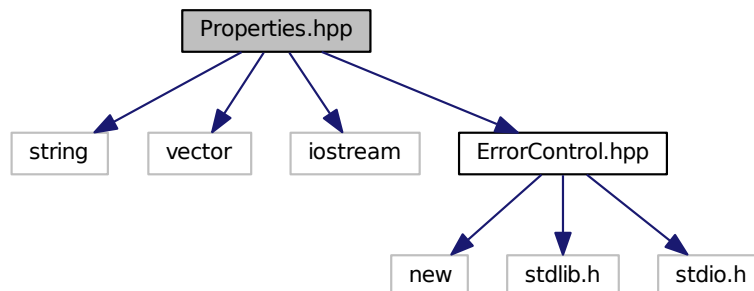
Include dependency graph for Properties.cpp:



## 8.86 Properties.hpp File Reference

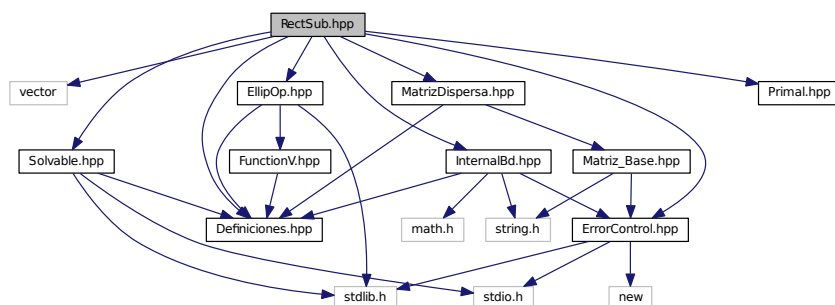
```
#include <string>
#include <vector>
#include <iostream>
#include "ErrorControl.hpp"
```

Include dependency graph for Properties.hpp:

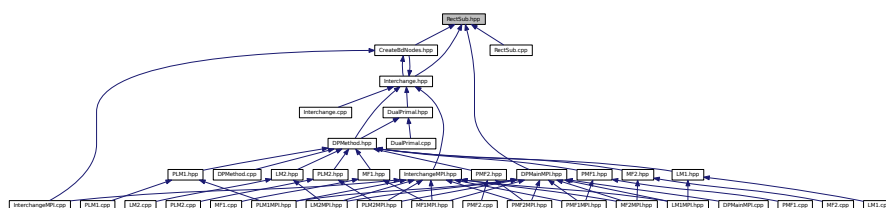




Include dependency graph for RectSub.hpp:



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



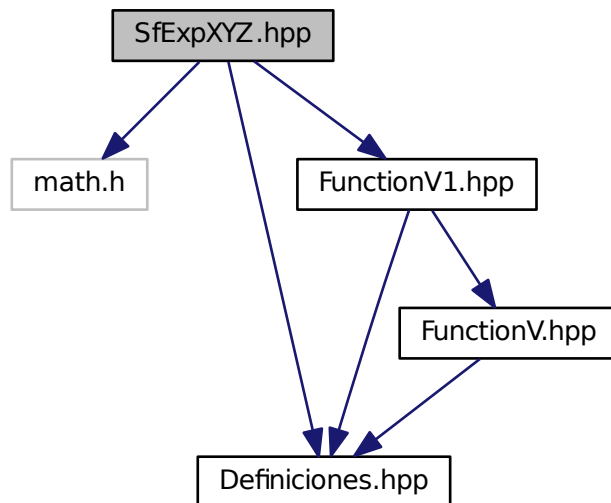
## Classes

- class RectSub

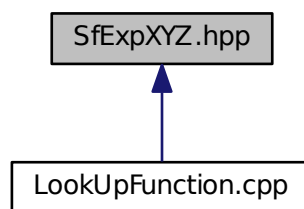
## 8.89 SfExpXYZ.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
```

Include dependency graph for SfExpXYZ.hpp:



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



## Classes

- class [SfExpXYZ](#)

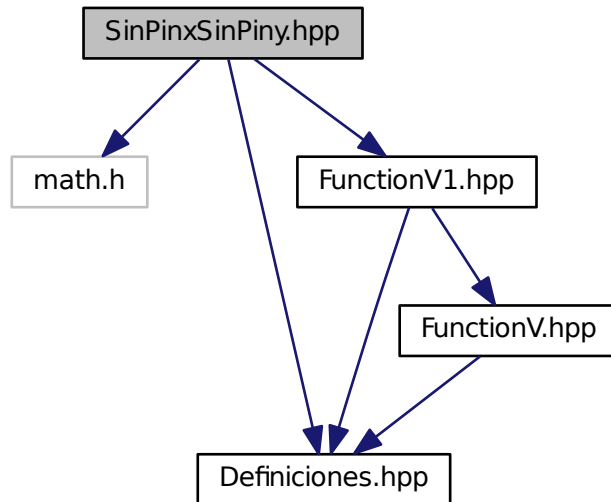
## 8.90 SinPinxSinPiny.hpp File Reference

```
#include <math.h>
```

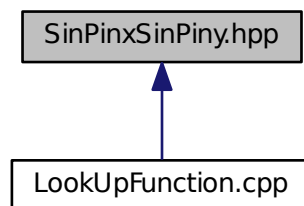
```
#include "Definiciones.hpp"
```

```
#include "FunctionV1.hpp"
```

Include dependency graph for SinPinxSinPiny.hpp:



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

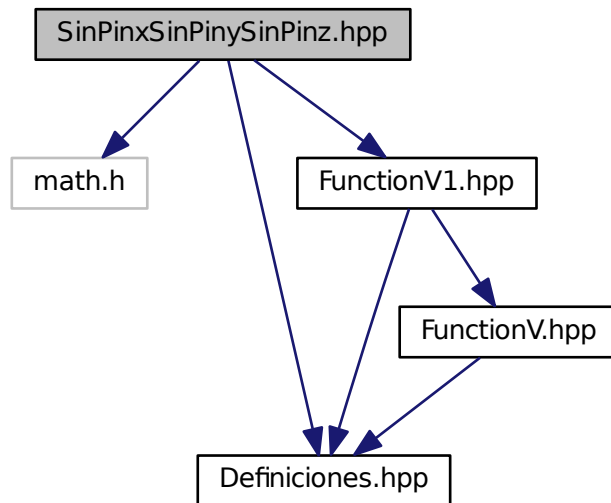


## Classes

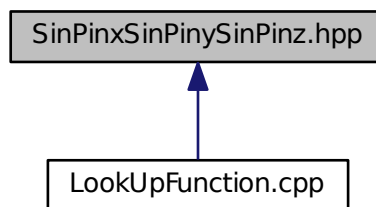
- class [SinPinxSinPiny](#)

## 8.91 SinPinxSinPinySinPinz.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for SinPinxSinPinySinPinz.hpp:
```



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



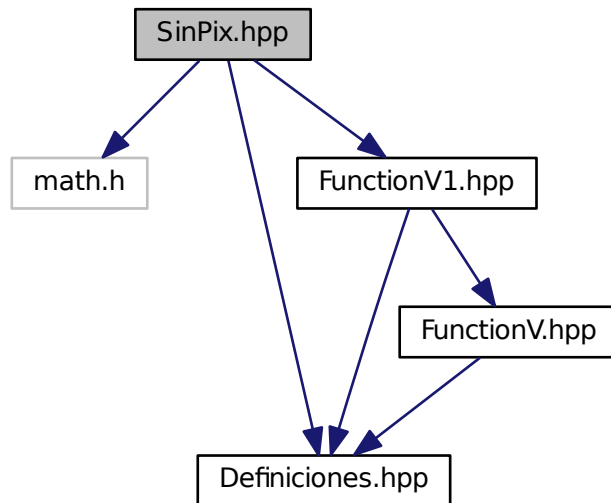
### Classes

- class [SinPinxSinPinySinPinz](#)

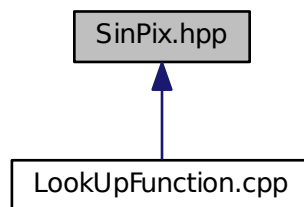


## 8.92 SinPix.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for SinPix.hpp:
```



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

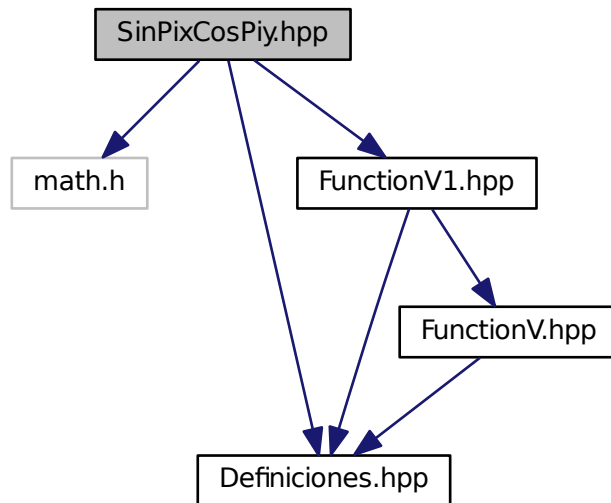


### Classes

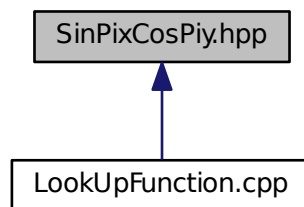
- class [SinPix](#)

### 8.93 SinPixCosPiy.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for SinPixCosPiy.hpp:
```



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

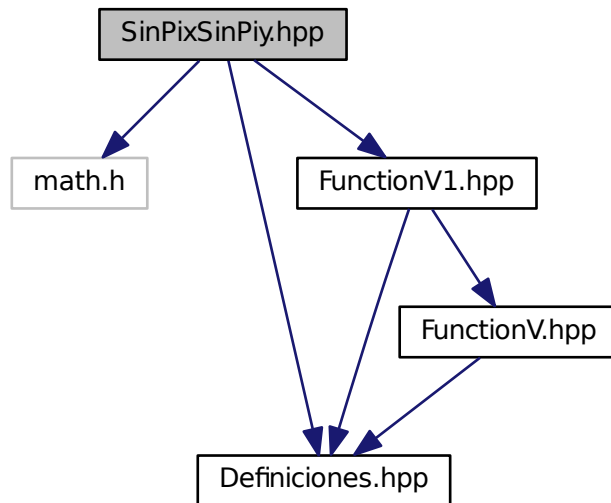


#### Classes

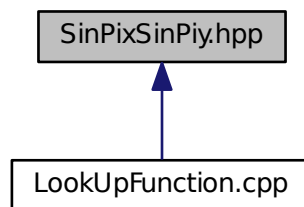
- class [SinPixCosPiy](#)

## 8.94 SinPixSinPiy.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for SinPixSinPiy.hpp:
```



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

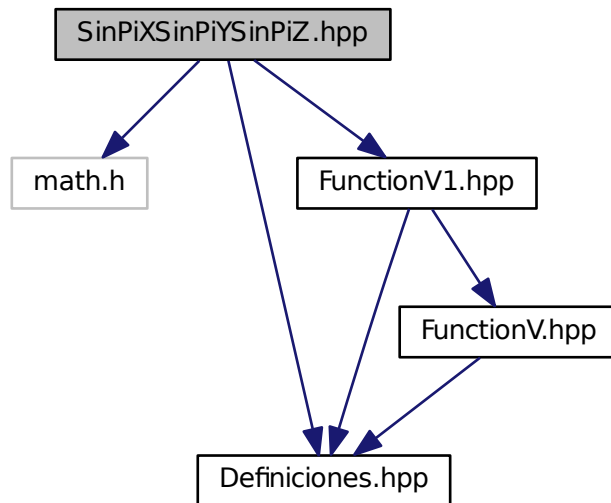


### Classes

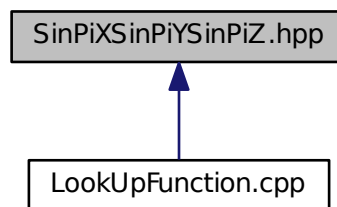
- class [SinPixSinPiy](#)

## 8.95 SinPiXSinPiYSinPiZ.hpp File Reference

```
#include <math.h>
#include "Definiciones.hpp"
#include "FunctionV1.hpp"
Include dependency graph for SinPiXSinPiYSinPiZ.hpp:
```



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



## Classes

- class [SinPiXSinPiYSinPiZ](#)

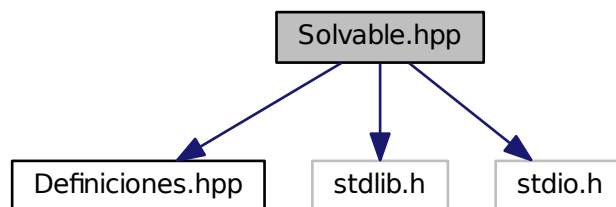
## 8.96 Solvable.hpp File Reference

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

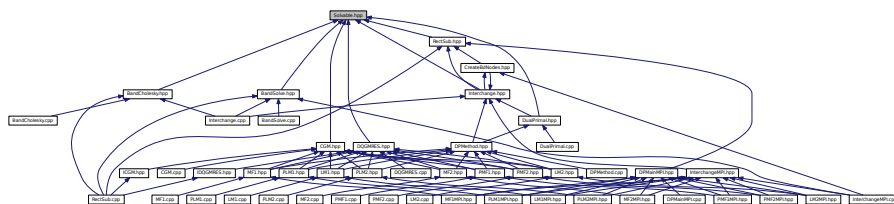
```
#include <stdlib.h>
```

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

Include dependency graph for Solvable.hpp:



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



## Classes

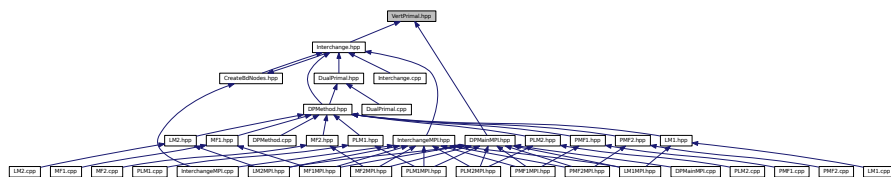
- class [Solvable](#)

## 8.97 VertEdgePrimal.hpp File Reference

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

Include dependency graph for V

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

- \_\_Double\_\_
  - Definiciones.hpp, [260](#)
- ~BandCholesky
  - BandCholesky, [20](#)
- ~BandSolve
  - BandSolve, [24](#)
- ~CGM
  - CGM, [29](#)
- ~CreateBdNodes
  - CreateBdNodes, [35](#)
- ~DPMainMPI
  - DPMainMPI, [52](#)
- ~DPMethod
  - DPMethod, [56](#)
- ~DQGMRES
  - DQGMRES, [61](#)
- ~DualPrimal
  - DualPrimal, [64](#)
- ~EllipOp
  - EllipOp, [67](#)
- ~EsquemaMEMPI
  - EsquemaMEMPI, [76](#)
- ~FunctionV
  - FunctionV, [96](#)
- ~HeapSort
  - HeapSort, [100](#)
- ~ICGM
  - ICGM, [103](#)
- ~IDQGMRES
  - IDQGMRES, [107](#)
- ~Interchange
  - Interchange, [112](#)
- ~InternalBd
  - InternalBd, [125](#)
- ~MatrizDispersa
  - MatrizDispersa, [150](#)
- ~Matriz\_Base
  - Matriz\_Base, [144](#)
- ~Primal
  - Primal, [206](#)
- ~RectSub
  - RectSub, [216](#)
- ~Solvable
  - Solvable, [243](#)

## A

- CGM, [30](#)
- a
  - Constant, [33](#)
  - DualPrimal, [64](#)
  - EllipOp, [67](#)
- AK
  - BandCholesky, [20](#)
  - BandSolve, [25](#)
  - MultBandSym, [168](#)
- AllPrimal, [15](#)
  - AllPrimal, [17](#)
  - isPrimal, [17](#)
  - name, [17](#)
- AllPrimal.hpp, [251](#)
- alpha
  - CGM, [30](#)
- analyticSolution
  - DPMethod, [56](#)
- applyOmega
  - DQGMRES, [61](#)
- asigna
  - MatrizDispersa, [150](#)
- asignaNombre
  - Matriz\_Base, [145](#)
- Ax
  - Interchange, [116](#)
- ax
  - Interchange, [116](#)
- Ay
  - Interchange, [116](#)
- ay
  - Interchange, [116](#)
- Az
  - Interchange, [116](#)
- az
  - Interchange, [116](#)
- b
  - EllipOp, [67](#)
- bFloat
  - RectSub, [218](#)
- Ban
  - Matriz\_Base, [146](#)
- BandCholesky, [17](#)
  - ~BandCholesky, [20](#)
- AK, [20](#)

- BandCholesky, 20
- bw, 20
- ce, 20
- clean, 20
- convertBand, 20
- factorLU, 20
- getIter, 20
- n, 21
- print, 20
- solve, 20
- BandCholesky.cpp, 252
- BandCholesky.hpp, 252
- BandSolve, 21
  - ~BandSolve, 24
  - AK, 25
  - BandSolve, 24
  - bw, 25
  - ce, 25
  - clean, 24
  - convertBand, 24
  - factorLU, 24
  - getIter, 24
  - n, 25
  - print, 24
  - solve, 25
- BandSolve.cpp, 254
- BandSolve.hpp, 254
- bd
  - InternalBd, 125
- bdAll
  - CreateBdNodes, 35
- bdDuals
  - CreateBdNodes, 35
- bdMap
  - RectSub, 218
- BdNode, 25
  - BdNode, 26
  - index, 26
  - mult, 26
  - node, 26
  - subd, 26
- BdNode.hpp, 256
- bdPrimals
  - CreateBdNodes, 35
- bdValues
  - Interchange, 116
- bds
  - Interchange, 116
- beta
  - CGM, 30
- bsym
  - RectSub, 218
- bw
  - BandCholesky, 20
  - BandSolve, 25
  - MultBandSym, 168
- bx
  - Interchange, 116
- by
  - Interchange, 116
- bz
  - Interchange, 116
- c
  - DPMainMPI, 52
  - EllipOp, 67
  - Interchange, 116
- CGM, 26
  - ~CGM, 29
  - A, 30
  - alpha, 30
  - beta, 30
  - CGM, 29
  - ce, 30
  - clean, 29
  - dotP, 30
  - eps, 30
  - gamma, 30
  - getIter, 29
  - inicializa, 29
  - lambda, 30
  - mu, 30
  - n, 30
  - nIter, 30
  - nMaxIter, 30
  - norm, 29
  - p, 30
  - r, 30
  - setEpsilon, 29
  - setMaxIter, 29
  - solve, 29
  - v, 30
- CGM.cpp, 256
- CGM.hpp, 256
- COEFICIENTES\_CONSTANTES
  - Definiciones.hpp, 260
- Ca
  - RectSub, 218
- calcSP
  - Interchange, 113
- calcValues
  - DualPrimal, 64
- calcula
  - Interchange, 113
  - InterchangeMPI, 121
- Cb
  - RectSub, 218
- Cc

- RectSub, 218
- ce
  - BandCholesky, 20
  - BandSolve, 25
  - CGM, 30
  - CreateBdNodes, 35
  - DPMMethod, 57
  - DQGMRES, 62
  - DualPrimal, 64
  - EsquemaMEMPI, 76
  - Interchange, 116
  - LookUpFunction, 141
  - Matriz\_Base, 146
  - Properties, 213
  - RectSub, 218
- clean
  - BandCholesky, 20
  - BandSolve, 24
  - CGM, 29
  - DPMMethod, 56
  - DQGMRES, 61
  - ICGM, 104
  - IDQGMRES, 107
  - LM1, 128
  - LM1MPI, 132
  - LM2, 135
  - LM2MPI, 139
  - MF1, 156
  - MF1MPI, 159
  - MF2, 162
  - MF2MPI, 165
  - PLM1, 181
  - PLM1MPI, 185
  - PLM2, 188
  - PLM2MPI, 192
  - PMF1, 195
  - PMF1MPI, 198
  - PMF2, 201
  - PMF2MPI, 204
  - Solvable, 243
- clear
  - Interchange, 113
  - InterchangeMPI, 121
  - RectSub, 216
- coef
  - RectSub, 218
- Col
  - Matriz\_Base, 146
- columnas
  - Matriz\_Base, 146
- compareTo
  - InternalBd, 125
- conditionalNumber
  - DPMMethod, 56
- Constant, 30
  - a, 33
  - Constant, 33
  - eval, 33
  - getVar, 33
  - setVar, 33
- Constant.hpp, 257
- convertBand
  - BandCholesky, 20
  - BandSolve, 24
- coord
  - InternalBd, 125
  - RectSub, 218
- coordN
  - RectSub, 218
- CreateBdNodes, 33
  - ~CreateBdNodes, 35
  - bdAll, 35
  - bdDuals, 35
  - bdPrimals, 35
  - ce, 35
  - CreateBdNodes, 35
  - dualMult, 35
  - hbd, 35
  - ibd, 35
  - ibdAll, 35
  - maxBd, 35
  - nD, 35
  - nDual, 35
  - nP, 35
  - nPrimal, 35
- CreateBdNodes.hpp, 258
- cs
  - DQGMRES, 62
- DIM\_VECTOR
  - Definiciones.hpp, 260
- DPMMainMPI, 49
  - ~DPMMainMPI, 52
  - c, 52
  - DPMMainMPI, 52
  - deleteInternalBd, 52
  - domain, 52
  - Esclavo, 52
  - f, 52
  - fc, 52
  - g, 52
  - gc, 52
  - hbd, 52
  - indl, 52
  - mesh, 52
  - method, 52
  - msa, 52
  - mss, 52

- nDim, [52](#)
- nOmega, [53](#)
- nta, [53](#)
- omegas, [53](#)
- one, [53](#)
- op, [53](#)
- prim, [53](#)
- primal, [53](#)
- sf, [53](#)
- sg, [53](#)
- swprint, [53](#)
- t1, [53](#)
- t2, [53](#)
- xnp, [53](#)
- zero, [53](#)
- DPMMainMPI.cpp, [265](#)
- DPMMainMPI.hpp, [266](#)
- DPMMethod, [53](#)
  - ~DPMMethod, [56](#)
  - analyticSolution, [56](#)
  - ce, [57](#)
  - clean, [56](#)
  - conditionalNumber, [56](#)
  - DPMMethod, [56](#)
  - dualp, [57](#)
  - epsilon, [57](#)
  - genInverse, [56](#)
  - getSize, [57](#)
  - iniInterchage, [57](#)
  - initialize, [57](#)
  - inter, [57](#)
  - nDim, [57](#)
  - nDual, [57](#)
  - nOmega, [57](#)
  - prCoord, [57](#)
  - print, [57](#)
  - printTime, [57](#)
  - printv, [57](#)
  - props, [57](#)
  - rhs, [57](#)
  - rhss, [58](#)
  - scr, [58](#)
  - solve, [57](#)
  - solver, [58](#)
  - time0, [58](#)
  - time1, [58](#)
  - time2, [58](#)
  - time3, [58](#)
  - u, [58](#)
- DPMMethod.cpp, [266](#)
- DPMMethod.hpp, [267](#)
- DQGMRES, [58](#)
  - ~DQGMRES, [61](#)
  - applyOmega, [61](#)
  - ce, [62](#)
  - clean, [61](#)
  - cs, [62](#)
  - DQGMRES, [61](#)
  - eps, [62](#)
  - getIter, [61](#)
  - gm, [62](#)
  - gm1, [62](#)
  - h, [62](#)
  - inicializa, [61](#)
  - k, [62](#)
  - k1, [62](#)
  - maxIter, [62](#)
  - mult, [62](#)
  - n, [62](#)
  - nIter, [62](#)
  - nMaxIter, [62](#)
  - p, [62](#)
  - q, [62](#)
  - setEpsilon, [61](#)
  - setMaxIter, [61](#)
  - solve, [61](#)
  - v, [62](#)
- DQGMRES.cpp, [268](#)
- DQGMRES.hpp, [268](#)
- DUAL
  - RectSub, [219](#)
- Definiciones.hpp, [259](#)
  - \_\_Double\_\_, [260](#)
  - COEFICIENTES\_CONSTANTES, [260](#)
  - DIM\_VECTOR, [260](#)
  - EPS\_EQUAL, [260](#)
  - EPSILON, [260](#)
  - EPSILON\_LOCAL, [260](#)
  - ldouble, [260](#)
  - NMAXITER, [260](#)
  - NMAXITER\_LOCAL, [260](#)
- deleteInternalBd
  - DPMMainMPI, [52](#)
- diff
  - Interchange, [113](#)
  - InterchangeMPI, [121](#)
  - RectSub, [216](#)
- diffValues
  - Interchange, [113](#)
  - InterchangeMPI, [121](#)
  - RectSub, [217](#)
- dim
  - FunctionV, [97](#)
- dimension
  - FunctionV, [96](#)
- Disc, [35](#)
  - Disc, [38](#)
  - eval, [38](#)

- getVar, [38](#)
  - setVar, [38](#)
  - var, [38](#)
- Disc.hpp, [261](#)
- Disc13, [38](#)
  - Disc13, [41](#)
  - eval, [41](#)
  - getVar, [41](#)
  - setVar, [41](#)
  - var, [41](#)
- Disc13.hpp, [262](#)
- Disc14, [41](#)
  - Disc14, [44](#)
  - eval, [44](#)
  - getVar, [44](#)
  - setVar, [44](#)
  - var, [44](#)
- Disc14.hpp, [263](#)
- Disc15, [44](#)
  - Disc15, [47](#)
  - eval, [47](#)
  - getVar, [47](#)
  - setVar, [47](#)
  - var, [47](#)
- Disc15.hpp, [264](#)
- domain
  - DPMMainMPI, [52](#)
  - Interchange, [116](#)
  - RectSub, [218](#)
- dot
  - DotProd, [48](#)
  - ICGM, [104](#)
  - LM1, [128](#)
  - LM2, [135](#)
  - MF1, [156](#)
  - MF2, [162](#)
  - PLM1, [181](#)
  - PLM2, [188](#)
  - PMF1, [195](#)
  - PMF2, [201](#)
- dotP
  - CGM, [30](#)
- DotProd, [47](#)
  - dot, [48](#)
- DotProd.hpp, [265](#)
- dp
  - InternalBd, [125](#)
- dualMult
  - CreateBdNodes, [35](#)
- DualPrimal, [62](#)
  - ~DualPrimal, [64](#)
  - a, [64](#)
  - calcValues, [64](#)
  - ce, [64](#)
  - DualPrimal, [64](#)
  - fromSubdomains, [64](#)
  - genMats, [64](#)
  - getNDual, [64](#)
  - inter, [64](#)
  - j, [64](#)
  - multS, [64](#)
  - nD, [64](#)
  - nDual, [64](#)
  - nP, [65](#)
  - nPrimal, [65](#)
  - SP, [65](#)
  - solveAPP, [64](#)
  - solveS, [64](#)
  - toSubdomains, [64](#)
  - XP, [65](#)
  - YP, [65](#)
- DualPrimal.cpp, [269](#)
- DualPrimal.hpp, [269](#)
- dualp
  - DPMMethod, [57](#)
- EDGE
  - Primal, [207](#)
  - RectSub, [219](#)
- EPS\_EQUAL
  - Definiciones.hpp, [260](#)
- EPSILON
  - Definiciones.hpp, [260](#)
- EPSILON\_LOCAL
  - Definiciones.hpp, [260](#)
- EllipOp, [65](#)
  - ~EllipOp, [67](#)
  - a, [67](#)
  - b, [67](#)
  - c, [67](#)
  - EllipOp, [67](#)
  - f, [68](#)
  - g, [68](#)
  - getA, [67](#)
  - getB, [67](#)
  - getC, [67](#)
  - getF, [67](#)
  - getG, [67](#)
  - isSymmetric, [67](#)
  - nDim, [68](#)
  - setF, [67](#)
  - setG, [67](#)
  - sol, [68](#)
- EllipOp.hpp, [270](#)
- eps
  - CGM, [30](#)
  - DQGMRES, [62](#)
- epsilon

- DPMMethod, 57
- equals
  - InternalBd, 125
- ErrorControl, 68
  - ErrorControl, 69, 70
  - fatalError, 70
  - memoryError, 70
  - nameClass, 72
  - nameClassFunct, 72
  - nameFunct, 72
  - nmClass, 72
  - nmFunction, 72
- ErrorControl.cpp, 271
- ErrorControl.hpp, 271
- Esclavo
  - DPMMainMPI, 52
- EsquemaMEMPI, 72
  - ~EsquemaMEMPI, 76
  - ce, 76
  - EsquemaMEMPI, 75
  - generaRepartoCarga, 76
  - id, 76
  - np, 76
  - npu, 77
  - numeroProcesadoresUsar, 76
  - reparteCargaTrabajo, 76
  - ta, 77
- EsquemaMEMPI.cpp, 272
- EsquemaMEMPI.hpp, 273
- eval
  - Constant, 33
  - Disc, 38
  - Disc13, 41
  - Disc14, 44
  - Disc15, 47
  - ExpVXY, 80
  - ExpVXYZ, 83
  - ExpX, 86
  - ExpXY, 89
  - ExpXYZ, 92
  - fExpXY, 95
  - FunctionV, 96
  - NSfExpXY, 175
  - NSfExpXYZ, 178
  - SfExpXYZ, 223
  - SinPiXSinPiYSinPiZ, 241
  - SinPinxSinPiny, 226
  - SinPinxSinPinySinPinz, 229
  - SinPix, 232
  - SinPixCosPiy, 235
  - SinPixSinPiy, 238
- ExpVXY, 77
  - eval, 80
  - ExpVXY, 80
  - getVar, 80
  - setVar, 80
  - var, 80
- ExpVXY.hpp, 273
- ExpVXYZ, 80
  - eval, 83
  - ExpVXYZ, 83
  - getVar, 83
  - setVar, 83
  - var, 83
- ExpVXYZ.hpp, 274
- ExpX, 83
  - eval, 86
  - ExpX, 86
  - getVar, 86
  - setVar, 86
  - var, 86
- ExpX.hpp, 276
- ExpXY, 86
  - eval, 89
  - ExpXY, 89
  - getVar, 89
  - setVar, 89
  - var, 89
- ExpXY.hpp, 277
- ExpXYZ, 89
  - eval, 92
  - ExpXYZ, 92
  - getVar, 92
  - setVar, 92
  - var, 92
- ExpXYZ.hpp, 278
- f
  - DPMMainMPI, 52
  - EllipOp, 68
  - Interchange, 116
  - LM1, 128
  - LM2, 135
  - PLM1, 181
  - PLM2, 188
- FACE
  - Primal, 207
  - RectSub, 219
- fExpXY, 92
  - eval, 95
  - fExpXY, 95
  - getVar, 95
  - setVar, 95
  - var, 95
- fExpXY.hpp, 279
- factorLU
  - BandCholesky, 20
  - BandSolve, 24



- fatalError
  - ErrorControl, 70
- fc
  - DPMMainMPI, 52
  - Interchange, 116
- fromSubdomains
  - DualPrimal, 64
  - Interchange, 113
  - InterchangeMPI, 121
- FunctionV, 95
  - ~FunctionV, 96
  - dim, 97
  - dimension, 96
  - eval, 96
  - FunctionV, 96
- FunctionV.hpp, 280
- FunctionV1, 97
  - getVar, 98
  - setVar, 98
- FunctionV1.hpp, 280
- g
  - DPMMainMPI, 52
  - EllipOp, 68
  - Interchange, 116
- gamma
  - CGM, 30
- gc
  - DPMMainMPI, 52
  - Interchange, 116
- genCoef
  - RectSub, 217
- genCoefVar
  - RectSub, 217
- genGeom
  - Interchange, 113
- genHeap
  - HeapSort, 100
- genInv
  - Interchange, 113
  - InterchangeMPI, 121
  - RectSub, 217
- genInverse
  - DPMMethod, 56
  - RectSub, 217
- genMats
  - DualPrimal, 64
- genNcoord
  - RectSub, 217
- genNtype
  - RectSub, 217
- generaRepartoCarga
  - EsquemaMEMPI, 76
- getA
  - EllipOp, 67
- getB
  - EllipOp, 67
- getBdSize
  - RectSub, 217
- getC
  - EllipOp, 67
- getCoord
  - RectSub, 217
- getCoordNode
  - Interchange, 113
  - InterchangeMPI, 121
  - RectSub, 217
- getDouble
  - PropDef, 210
- getF
  - EllipOp, 67
  - LookUpFunction, 141
- getG
  - EllipOp, 67
- getInt
  - PropDef, 210
- getInternalBd
  - InterchangeMPI, 121
  - RectSub, 217
- getInv
  - RectSub, 217
- getIter
  - BandCholesky, 20
  - BandSolve, 24
  - CGM, 29
  - DQGMRES, 61
  - Solvable, 244
- getMaxBdSize
  - Interchange, 113
  - InterchangeMPI, 121
- getND
  - Interchange, 114
- getNDual
  - DualPrimal, 64
- getNDuals
  - Interchange, 114
- getNP
  - Interchange, 114
  - RectSub, 217
- getNPPrimals
  - Interchange, 114
- getName
  - Solvable, 244
- getNtype
  - Interchange, 114
  - InterchangeMPI, 121
  - RectSub, 217
- getPrimals

- Interchange, 114
- InterchangeMPI, 121
- RectSub, 217
- getProperty
  - Properties, 213
- getSize
  - DPMMethod, 57
  - ICGM, 104
  - IDQGMRES, 107
  - MultBandSym, 167
  - MultOp, 169
- getString
  - PropDef, 210
- getValue
  - Interchange, 114
  - InterchangeMPI, 122
  - RectSub, 217
- getValues
  - RectSub, 217
- getVar
  - Constant, 33
  - Disc, 38
  - Disc13, 41
  - Disc14, 44
  - Disc15, 47
  - ExpVXY, 80
  - ExpVXYZ, 83
  - ExpX, 86
  - ExpXY, 89
  - ExpXYZ, 92
  - fExpXY, 95
  - FunctionV1, 98
  - NSfExpXY, 175
  - NSfExpXYZ, 178
  - SfExpXYZ, 223
  - SinPiXSinPiYSinPiZ, 241
  - SinPinxSinPiny, 226
  - SinPinxSinPinySinPinz, 229
  - SinPix, 232
  - SinPixCosPiy, 235
  - SinPixSinPiy, 238
- getnDim
  - Interchange, 114
- getnOmega
  - Interchange, 114
- getval
  - InternalBd, 125
- gm
  - DQGMRES, 62
- gm1
  - DQGMRES, 62
- h
  - DQGMRES, 62
- RectSub, 219
- hbd
  - CreateBdNodes, 35
  - DPMMainMPI, 52
- HeapSort, 99
  - ~HeapSort, 100
  - genHeap, 100
  - HeapSort, 100
  - n, 100
  - r, 100
  - rr, 100
  - siftup, 100
  - sort, 100
  - swap, 100
- HeapSort.hpp, 281
- hfac
  - RectSub, 219
- i
  - ICGM, 104
- ICGM, 100
  - ~ICGM, 103
  - clean, 104
  - dot, 104
  - getSize, 104
  - i, 104
  - ICGM, 103
  - M, 104
  - multOp, 104
  - val, 104
- ICGM.hpp, 282
- IDQGMRES, 104
  - ~IDQGMRES, 107
  - clean, 107
  - getSize, 107
  - IDQGMRES, 107
  - M, 108
  - multOp, 108
- IDQGMRES.hpp, 283
- INTBD
  - Primal, 207
  - RectSub, 219
- INTERIOR
  - Primal, 207
  - RectSub, 219
- ibd
  - CreateBdNodes, 35
- ibdAll
  - CreateBdNodes, 35
- id
  - EsquemaMEMPI, 76
  - RectSub, 219
- index
  - BdNode, 26

- indl
  - DPMainMPI, 52
  - InterchangeMPI, 123
- iniInterchage
  - DPMMethod, 57
  - LM1MPI, 132
  - LM2MPI, 139
  - MF1MPI, 159
  - MF2MPI, 165
  - PLM1MPI, 185
  - PLM2MPI, 192
  - PMF1MPI, 198
  - PMF2MPI, 204
- inicializa
  - CGM, 29
  - DQGMRES, 61
  - MatrizDispersa, 151
- initialize
  - DPMMethod, 57
  - Interchange, 114
- inter
  - DPMMethod, 57
  - DualPrimal, 64
- Interchange, 108
  - ~Interchange, 112
  - Ax, 116
  - ax, 116
  - Ay, 116
  - ay, 116
  - Az, 116
  - az, 116
  - bdValues, 116
  - bds, 116
  - bx, 116
  - by, 116
  - bz, 116
  - c, 116
  - calcSP, 113
  - calcula, 113
  - ce, 116
  - clear, 113
  - diff, 113
  - diffValues, 113
  - domain, 116
  - f, 116
  - fc, 116
  - fromSubdomains, 113
  - g, 116
  - gc, 116
  - genGeom, 113
  - genInv, 113
  - getCoordNode, 113
  - getMaxBdSize, 113
  - getND, 114
  - getNDuals, 114
  - getNP, 114
  - getNPrimals, 114
  - getNtype, 114
  - getPrimals, 114
  - getValue, 114
  - getnDim, 114
  - getnOmega, 114
  - initialize, 114
  - Interchange, 112
  - inverse, 114
  - knownValues, 114
  - mesh, 116
  - method, 116
  - multOp, 114, 115
  - nDim, 116
  - nOmega, 116
  - omegas, 116
  - one, 116
  - op, 116
  - pbdValues, 115
  - prim, 117
  - primal, 117
  - print, 115
  - props, 117
  - rbdValues, 115
  - rhs, 115
  - sbdValues, 115
  - setNtype, 115
  - setPrimals, 115
  - setValue, 115
  - sf, 117
  - sg, 117
  - swprint, 117
  - toSubdomains, 115
  - zero, 117
- Interchange.cpp, 284
- Interchange.hpp, 285
- InterchangeMPI, 117
  - calcula, 121
  - clear, 121
  - diff, 121
  - diffValues, 121
  - fromSubdomains, 121
  - genInv, 121
  - getCoordNode, 121
  - getInternalBd, 121
  - getMaxBdSize, 121
  - getNtype, 121
  - getPrimals, 121
  - getValue, 122
  - indl, 123
  - InterchangeMPI, 120
  - inverse, 122

- knownValues, [122](#)
- ME, [123](#)
- msa, [123](#)
- mss, [123](#)
- multOp, [122](#)
- print, [122](#)
- rhs, [122](#)
- setNtype, [122](#)
- setPrimals, [122](#)
- toSubdomains, [122](#)
- xnp, [123](#)
- InterchangeMPI.cpp, [286](#)
- InterchangeMPI.hpp, [286](#)
- InternalBd, [123](#)
  - ~InternalBd, [125](#)
  - bd, [125](#)
  - compareTo, [125](#)
  - coord, [125](#)
  - dp, [125](#)
  - equals, [125](#)
  - getval, [125](#)
  - InternalBd, [125](#)
  - nDim, [125](#)
  - node, [125](#)
  - rbd, [125](#)
  - rcoord, [125](#)
  - rdp, [125](#)
  - rnode, [125](#)
  - rsubd, [125](#)
  - subd, [125](#)
- InternalBd.hpp, [287](#)
- inv
  - RectSub, [219](#)
- inverse
  - Interchange, [114](#)
  - InterchangeMPI, [122](#)
  - RectSub, [217](#)
- isDual
  - RectSub, [217](#)
- isFloat
  - RectSub, [217](#)
- isIntBd
  - RectSub, [217](#)
- isInterior
  - RectSub, [217](#)
- isKnown
  - RectSub, [217](#)
- isPrimal
  - AllPrimal, [17](#)
  - NoPrimal, [172](#)
  - Primal, [207](#)
  - RectSub, [217](#)
  - VertEdgePrimal, [247](#)
  - VertPrimal, [250](#)
- isSymmetric
  - EllipOp, [67](#)
- isVertex
  - RectSub, [218](#)
- J
  - MatrizDispersa, [153](#)
- j
  - DualPrimal, [64](#)
- k
  - DQGMRES, [62](#)
- k1
  - DQGMRES, [62](#)
- KNOWN
  - Primal, [207](#)
  - RectSub, [219](#)
- keys
  - Properties, [213](#)
- knownValues
  - Interchange, [114](#)
  - InterchangeMPI, [122](#)
  - RectSub, [218](#)
- LM1, [126](#)
  - clean, [128](#)
  - dot, [128](#)
  - f, [128](#)
  - LM1, [128](#)
  - multOp, [128](#)
  - op, [128](#)
  - rhs, [128](#)
  - solve, [128](#)
  - uf, [128](#)
  - up, [128](#)
- LM1.cpp, [287](#)
- LM1.hpp, [288](#)
- LM1MPI, [129](#)
  - clean, [132](#)
  - iniInterchange, [132](#)
  - LM1MPI, [132](#)
  - solvePar, [132](#)
- LM1MPI.hpp, [289](#)
- LM2, [132](#)
  - clean, [135](#)
  - dot, [135](#)
  - f, [135](#)
  - LM2, [135](#)
  - multOp, [135](#)
  - op, [135](#)
  - rhs, [135](#)
  - solve, [135](#)
  - uf, [135](#)
  - ug, [135](#)
  - up, [136](#)

- LM2.cpp, [289](#)
- LM2.hpp, [289](#)
- LM2MPI, [136](#)
  - clean, [139](#)
  - iniInterchage, [139](#)
  - LM2MPI, [139](#)
  - solvePar, [139](#)
- LM2MPI.hpp, [290](#)
- lambda
  - CGM, [30](#)
- ldouble
  - Definiciones.hpp, [260](#)
- list
  - Properties, [213](#)
- load
  - Properties, [213](#)
- LookUpFunction, [139](#)
  - ce, [141](#)
  - getF, [141](#)
- LookUpFunction.cpp, [291](#)
- LookUpFunction.hpp, [291](#)
- M
  - ICGM, [104](#)
  - IDQGMRES, [108](#)
  - MatrizDispersa, [153](#)
  - RectSub, [219](#)
- M1
  - RectSub, [219](#)
- ME
  - InterchangeMPI, [123](#)
- MF1, [153](#)
  - clean, [156](#)
  - dot, [156](#)
  - MF1, [156](#)
  - multOp, [156](#)
  - op, [156](#)
  - rhs, [156](#)
  - solve, [156](#)
- MF1.cpp, [295](#)
- MF1.hpp, [296](#)
- MF1MPI, [156](#)
  - clean, [159](#)
  - iniInterchage, [159](#)
  - MF1MPI, [159](#)
  - solvePar, [159](#)
- MF1MPI.hpp, [297](#)
- MF2, [159](#)
  - clean, [162](#)
  - dot, [162](#)
  - MF2, [162](#)
  - multOp, [162](#)
  - op, [162](#)
  - rhs, [162](#)
  - solve, [162](#)
  - up, [162](#)
- MF2.cpp, [297](#)
- MF2.hpp, [297](#)
- MF2MPI, [163](#)
  - clean, [165](#)
  - iniInterchage, [165](#)
  - MF2MPI, [165](#)
  - solvePar, [165](#)
- MF2MPI.hpp, [298](#)
- main.hpp, [292](#)
- mapFull
  - RectSub, [219](#)
- mapInt
  - RectSub, [219](#)
- Matriz\_Base, [141](#)
  - ~Matriz\_Base, [144](#)
  - asignaNombre, [145](#)
  - Ban, [146](#)
  - ce, [146](#)
  - Col, [146](#)
  - columnas, [146](#)
  - Matriz\_Base, [144](#)
  - Nmb, [146](#)
  - Ren, [146](#)
  - renglones, [146](#)
- Matriz\_Base.hpp, [292](#)
- MatrizDispersa, [147](#)
  - ~MatrizDispersa, [150](#)
  - asigna, [150](#)
  - inicializa, [151](#)
  - J, [153](#)
  - M, [153](#)
  - MatrizDispersa, [150](#)
  - multiplica, [151](#)
  - retorna, [151](#)
  - retornaNumeroColumna, [151](#)
  - retornaNumeroColumnasBanda, [151](#)
  - retornaValorColumna, [152](#)
  - solicitaMemoria, [152](#)
  - tamanoBanda, [152](#)
  - visualiza, [152](#)
  - visualizaMatricesInternas, [152](#)
- MatrizDispersa.cpp, [293](#)
- MatrizDispersa.hpp, [294](#)
- maxBd
  - CreateBdNodes, [35](#)
- maxIter
  - DQGMRES, [62](#)
- memoryError
  - ErrorControl, [70](#)
- mesh
  - DPMainMPI, [52](#)
  - Interchange, [116](#)

- RectSub, [219](#)
- method
  - DPMainMPI, [52](#)
  - Interchange, [116](#)
- msa
  - DPMainMPI, [52](#)
  - InterchangeMPI, [123](#)
- mss
  - DPMainMPI, [52](#)
  - InterchangeMPI, [123](#)
- mu
  - CGM, [30](#)
- mult
  - BdNode, [26](#)
  - DQGMRES, [62](#)
- MultBandSym, [165](#)
  - AK, [168](#)
  - bw, [168](#)
  - getSize, [167](#)
  - MultBandSym, [167](#)
  - multOp, [168](#)
  - n, [168](#)
- MultBandSym.cpp, [299](#)
- MultBandSym.hpp, [299](#)
- MultOp, [168](#)
  - getSize, [169](#)
  - multOp, [169](#)
- multOp
  - ICGM, [104](#)
  - IDQGMRES, [108](#)
  - Interchange, [114](#), [115](#)
  - InterchangeMPI, [122](#)
  - LM1, [128](#)
  - LM2, [135](#)
  - MF1, [156](#)
  - MF2, [162](#)
  - MultBandSym, [168](#)
  - MultOp, [169](#)
  - PLM1, [181](#)
  - PLM2, [188](#)
  - PMF1, [195](#)
  - PMF2, [201](#)
  - RectSub, [218](#)
- MultOp.hpp, [300](#)
- multS
  - DualPrimal, [64](#)
- multiplica
  - MatrizDispersa, [151](#)
- N
  - RectSub, [219](#)
- n
  - BandCholesky, [21](#)
  - BandSolve, [25](#)
  - CGM, [30](#)
  - DQGMRES, [62](#)
  - HeapSort, [100](#)
  - MultBandSym, [168](#)
  - SinPinxSinPiny, [226](#)
  - SinPinxSinPinySinPinz, [229](#)
- nBd
  - RectSub, [219](#)
- nD
  - CreateBdNodes, [35](#)
  - DualPrimal, [64](#)
- nDim
  - DPMainMPI, [52](#)
  - DPMMethod, [57](#)
  - EllipOp, [68](#)
  - Interchange, [116](#)
  - InternalBd, [125](#)
  - RectSub, [219](#)
- nDual
  - CreateBdNodes, [35](#)
  - DPMMethod, [57](#)
  - DualPrimal, [64](#)
- nFull
  - RectSub, [219](#)
- nInt
  - RectSub, [219](#)
- nIter
  - CGM, [30](#)
  - DQGMRES, [62](#)
- NMAXITER
  - Definiciones.hpp, [260](#)
- NMAXITER\_LOCAL
  - Definiciones.hpp, [260](#)
- nMaxIter
  - CGM, [30](#)
  - DQGMRES, [62](#)
- nOmega
  - DPMainMPI, [53](#)
  - DPMMethod, [57](#)
  - Interchange, [116](#)
- nP
  - CreateBdNodes, [35](#)
  - DualPrimal, [65](#)
- nPrimal
  - CreateBdNodes, [35](#)
  - DualPrimal, [65](#)
- NSfExpXY, [172](#)
  - eval, [175](#)
  - getVar, [175](#)
  - NSfExpXY, [175](#)
  - setVar, [175](#)
  - var, [175](#)
- NSfExpXY.hpp, [302](#)
- NSfExpXYZ, [175](#)

- eval, [178](#)
- getVar, [178](#)
- NSfExpXYZ, [178](#)
- setVar, [178](#)
- var, [178](#)
- NSfExpXYZ.hpp, [303](#)
- name
  - AllPrimal, [17](#)
  - NoPrimal, [172](#)
  - Solvable, [244](#)
  - VertEdgePrimal, [247](#)
  - VertPrimal, [250](#)
- nameClass
  - ErrorControl, [72](#)
- nameClassFunct
  - ErrorControl, [72](#)
- nameFunct
  - ErrorControl, [72](#)
- nmClass
  - ErrorControl, [72](#)
- nmFunction
  - ErrorControl, [72](#)
- Nmb
  - Matriz\_Base, [146](#)
- NoPrimal, [169](#)
  - isPrimal, [172](#)
  - name, [172](#)
  - NoPrimal, [172](#)
- NoPrimal.hpp, [301](#)
- node
  - BdNode, [26](#)
  - InternalBd, [125](#)
- nodeType
  - RectSub, [218](#)
- norm
  - CGM, [29](#)
- np
  - EsquemaMEMPI, [76](#)
  - RectSub, [219](#)
- npu
  - EsquemaMEMPI, [77](#)
- nta
  - DPMMainMPI, [53](#)
- ntype
  - RectSub, [219](#)
- numeroProcesadoresUsar
  - EsquemaMEMPI, [76](#)
- omegas
  - DPMMainMPI, [53](#)
  - Interchange, [116](#)
- one
  - DPMMainMPI, [53](#)
  - Interchange, [116](#)
- op
  - DPMMainMPI, [53](#)
  - Interchange, [116](#)
  - LM1, [128](#)
  - LM2, [135](#)
  - MF1, [156](#)
  - MF2, [162](#)
  - PLM1, [181](#)
  - PLM2, [188](#)
  - PMF1, [195](#)
  - PMF2, [201](#)
  - RectSub, [219](#)
- p
  - CGM, [30](#)
  - DQGMRES, [62](#)
- PLM1, [178](#)
  - clean, [181](#)
  - dot, [181](#)
  - f, [181](#)
  - multOp, [181](#)
  - op, [181](#)
  - PLM1, [181](#)
  - rhs, [181](#)
  - solve, [181](#)
  - uf, [181](#)
  - up, [181](#)
- PLM1.cpp, [304](#)
- PLM1.hpp, [304](#)
- PLM1MPI, [182](#)
  - clean, [185](#)
  - iniInterchage, [185](#)
  - PLM1MPI, [185](#)
  - solvePar, [185](#)
- PLM1MPI.hpp, [305](#)
- PLM2, [185](#)
  - clean, [188](#)
  - dot, [188](#)
  - f, [188](#)
  - multOp, [188](#)
  - op, [188](#)
  - PLM2, [188](#)
  - rhs, [188](#)
  - solve, [188](#)
  - uf, [188](#)
  - ug, [188](#)
  - up, [189](#)
- PLM2.cpp, [306](#)
- PLM2.hpp, [306](#)
- PLM2MPI, [189](#)
  - clean, [192](#)
  - iniInterchage, [192](#)
  - PLM2MPI, [192](#)
  - solvePar, [192](#)

- PLM2MPI.hpp, 307
- PMF1, 192
  - clean, 195
  - dot, 195
  - multOp, 195
  - op, 195
  - PMF1, 195
  - rhs, 195
  - solve, 195
- PMF1.cpp, 307
- PMF1.hpp, 308
- PMF1MPI, 195
  - clean, 198
  - iniInterchage, 198
  - PMF1MPI, 198
  - solvePar, 198
- PMF1MPI.hpp, 309
- PMF2, 198
  - clean, 201
  - dot, 201
  - multOp, 201
  - op, 201
  - PMF2, 201
  - rhs, 201
  - solve, 201
  - uf, 201
  - up, 201
- PMF2.cpp, 309
- PMF2.hpp, 309
- PMF2MPI, 202
  - clean, 204
  - iniInterchage, 204
  - PMF2MPI, 204
  - solvePar, 204
- PMF2MPI.hpp, 310
- PRIMAL
  - Primal, 207
  - RectSub, 219
- parse
  - PropDef, 210
- pbdValues
  - Interchange, 115
- prCoord
  - DPMMethod, 57
- prim
  - DPMMainMPI, 53
  - Interchange, 117
- Primal, 204
  - ~Primal, 206
  - EDGE, 207
  - FACE, 207
  - INTBD, 207
  - INTERIOR, 207
  - isPrimal, 207
  - KNOWN, 207
  - PRIMAL, 207
  - Primal, 206
  - VERTEX, 207
- primal
  - DPMMainMPI, 53
  - Interchange, 117
- Primal.hpp, 311
- print
  - BandCholesky, 20
  - BandSolve, 24
  - DPMMethod, 57
  - Interchange, 115
  - InterchangeMPI, 122
  - RectSub, 218
- printMat
  - RectSub, 218
- printMult
  - RectSub, 218
- printTime
  - DPMMethod, 57
- printv
  - DPMMethod, 57
- PropDef, 207
  - getDouble, 210
  - getInt, 210
  - getString, 210
  - parse, 210
  - PropDef, 210
- PropDef.cpp, 311
- PropDef.hpp, 312
- Properties, 210
  - ce, 213
  - getProperty, 213
  - keys, 213
  - list, 213
  - load, 213
  - Properties, 213
  - setProperty, 213
  - vals, 213
- Properties.cpp, 312
- Properties.hpp, 313
- props
  - DPMMethod, 57
  - Interchange, 117
- q
  - DQGMRES, 62
- r
  - CGM, 30
  - HeapSort, 100
- rbd
  - InternalBd, 125
- rbdValues



- Interchange, [115](#)
- rcoord
  - InternalBd, [125](#)
- rdp
  - InternalBd, [125](#)
- RectSub, [213](#)
  - ~RectSub, [216](#)
  - bFloat, [218](#)
  - bdMap, [218](#)
  - bsym, [218](#)
  - Ca, [218](#)
  - Cb, [218](#)
  - Cc, [218](#)
  - ce, [218](#)
  - clear, [216](#)
  - coef, [218](#)
  - coord, [218](#)
  - coordN, [218](#)
  - DUAL, [219](#)
  - diff, [216](#)
  - diffValues, [217](#)
  - domain, [218](#)
  - EDGE, [219](#)
  - FACE, [219](#)
  - genCoef, [217](#)
  - genCoefVar, [217](#)
  - genInv, [217](#)
  - genInverse, [217](#)
  - genNcoord, [217](#)
  - genNtype, [217](#)
  - getBdSize, [217](#)
  - getCoord, [217](#)
  - getCoordNode, [217](#)
  - getInternalBd, [217](#)
  - getInv, [217](#)
  - getNP, [217](#)
  - getNtype, [217](#)
  - getPrimals, [217](#)
  - getValue, [217](#)
  - getValues, [217](#)
  - h, [219](#)
  - hfac, [219](#)
  - INTBD, [219](#)
  - INTERIOR, [219](#)
  - id, [219](#)
  - inv, [219](#)
  - inverse, [217](#)
  - isDual, [217](#)
  - isFloat, [217](#)
  - isIntBd, [217](#)
  - isInterior, [217](#)
  - isKnown, [217](#)
  - isPrimal, [217](#)
  - isVertex, [218](#)
  - KNOWN, [219](#)
  - knownValues, [218](#)
  - M, [219](#)
  - M1, [219](#)
  - mapFull, [219](#)
  - mapInt, [219](#)
  - mesh, [219](#)
  - multOp, [218](#)
  - N, [219](#)
  - nBd, [219](#)
  - nDim, [219](#)
  - nFull, [219](#)
  - nInt, [219](#)
  - nodeType, [218](#)
  - np, [219](#)
  - ntype, [219](#)
  - op, [219](#)
  - PRIMAL, [219](#)
  - print, [218](#)
  - printMat, [218](#)
  - printMult, [218](#)
  - RectSub, [216](#)
  - rhs, [218](#)
  - scr, [219](#)
  - setNtype, [218](#)
  - setPrimals, [218](#)
  - setValue, [218](#)
  - setValues, [218](#)
  - VERTEX, [219](#)
  - X, [220](#)
  - x, [220](#)
  - Y, [220](#)
  - RectSub.cpp, [314](#)
  - RectSub.hpp, [314](#)
  - Ren
    - Matriz\_Base, [146](#)
  - renglones
    - Matriz\_Base, [146](#)
  - reparteCargaTrabajo
    - EsquemaMEMPI, [76](#)
  - retorna
    - MatrizDispersa, [151](#)
  - retornaNumeroColumna
    - MatrizDispersa, [151](#)
  - retornaNumeroColumnasBanda
    - MatrizDispersa, [151](#)
  - retornaValorColumna
    - MatrizDispersa, [152](#)
  - rhs
    - DPMMethod, [57](#)
    - Interchange, [115](#)
    - InterchangeMPI, [122](#)
    - LM1, [128](#)
    - LM2, [135](#)

- MF1, [156](#)
- MF2, [162](#)
- PLM1, [181](#)
- PLM2, [188](#)
- PMF1, [195](#)
- PMF2, [201](#)
- RectSub, [218](#)
- rhss
  - DPMMethod, [58](#)
- rnode
  - InternalBd, [125](#)
- rr
  - HeapSort, [100](#)
- rsubd
  - InternalBd, [125](#)
- SP
  - DualPrimal, [65](#)
- sbdValues
  - Interchange, [115](#)
- scr
  - DPMMethod, [58](#)
  - RectSub, [219](#)
- setEpsilon
  - CGM, [29](#)
  - DQGMRES, [61](#)
- setF
  - EllipOp, [67](#)
- setG
  - EllipOp, [67](#)
- setMaxIter
  - CGM, [29](#)
  - DQGMRES, [61](#)
- setNtype
  - Interchange, [115](#)
  - InterchangeMPI, [122](#)
  - RectSub, [218](#)
- setPrimals
  - Interchange, [115](#)
  - InterchangeMPI, [122](#)
  - RectSub, [218](#)
- setProperty
  - Properties, [213](#)
- setValue
  - Interchange, [115](#)
  - RectSub, [218](#)
- setValues
  - RectSub, [218](#)
- setVar
  - Constant, [33](#)
  - Disc, [38](#)
  - Disc13, [41](#)
  - Disc14, [44](#)
  - Disc15, [47](#)
  - ExpVXY, [80](#)
  - ExpVXYZ, [83](#)
  - ExpX, [86](#)
  - ExpXY, [89](#)
  - ExpXYZ, [92](#)
  - fExpXY, [95](#)
  - FunctionV1, [98](#)
  - NSfExpXY, [175](#)
  - NSfExpXYZ, [178](#)
  - SfExpXYZ, [223](#)
  - SinPiXSinPiYSinPiZ, [241](#)
  - SinPinxSinPiny, [226](#)
  - SinPinxSinPinySinPinz, [229](#)
  - SinPix, [232](#)
  - SinPixCosPiy, [235](#)
  - SinPixSinPiy, [238](#)
- sf
  - DPMainMPI, [53](#)
  - Interchange, [117](#)
- SfExpXYZ, [220](#)
  - eval, [223](#)
  - getVar, [223](#)
  - setVar, [223](#)
  - SfExpXYZ, [223](#)
  - var, [223](#)
- SfExpXYZ.hpp, [315](#)
- sg
  - DPMainMPI, [53](#)
  - Interchange, [117](#)
- siftup
  - HeapSort, [100](#)
- SinPiXSinPiYSinPiZ, [238](#)
  - eval, [241](#)
  - getVar, [241](#)
  - setVar, [241](#)
  - SinPiXSinPiYSinPiZ, [241](#)
  - var, [241](#)
- SinPiXSinPiYSinPiZ.hpp, [322](#)
- SinPinxSinPiny, [223](#)
  - eval, [226](#)
  - getVar, [226](#)
  - n, [226](#)
  - setVar, [226](#)
  - SinPinxSinPiny, [226](#)
  - var, [226](#)
- SinPinxSinPiny.hpp, [316](#)
- SinPinxSinPinySinPinz, [226](#)
  - eval, [229](#)
  - getVar, [229](#)
  - n, [229](#)
  - setVar, [229](#)
  - SinPinxSinPinySinPinz, [229](#)
  - var, [229](#)
- SinPinxSinPinySinPinz.hpp, [318](#)

- SinPix, [229](#)
  - eval, [232](#)
  - getVar, [232](#)
  - setVar, [232](#)
  - SinPix, [232](#)
  - var, [232](#)
- SinPix.hpp, [319](#)
- SinPixCosPiy, [232](#)
  - eval, [235](#)
  - getVar, [235](#)
  - setVar, [235](#)
  - SinPixCosPiy, [235](#)
  - var, [235](#)
- SinPixCosPiy.hpp, [320](#)
- SinPixSinPiy, [235](#)
  - eval, [238](#)
  - getVar, [238](#)
  - setVar, [238](#)
  - SinPixSinPiy, [238](#)
  - var, [238](#)
- SinPixSinPiy.hpp, [321](#)
- sol
  - EllipOp, [68](#)
- solicitaMemoria
  - MatrizDispersa, [152](#)
- Solvable, [241](#)
  - ~Solvable, [243](#)
  - clean, [243](#)
  - getIter, [244](#)
  - getName, [244](#)
  - name, [244](#)
  - Solvable, [243](#)
  - solve, [244](#)
- Solvable.hpp, [323](#)
- solve
  - BandCholesky, [20](#)
  - BandSolve, [25](#)
  - CGM, [29](#)
  - DPMMethod, [57](#)
  - DQGMRES, [61](#)
  - LM1, [128](#)
  - LM2, [135](#)
  - MF1, [156](#)
  - MF2, [162](#)
  - PLM1, [181](#)
  - PLM2, [188](#)
  - PMF1, [195](#)
  - PMF2, [201](#)
  - Solvable, [244](#)
- solveAPP
  - DualPrimal, [64](#)
- solvePar
  - LM1MPI, [132](#)
  - LM2MPI, [139](#)
- MF1MPI, [159](#)
- MF2MPI, [165](#)
- PLM1MPI, [185](#)
- PLM2MPI, [192](#)
- PMF1MPI, [198](#)
- PMF2MPI, [204](#)
- solveS
  - DualPrimal, [64](#)
- solver
  - DPMMethod, [58](#)
- sort
  - HeapSort, [100](#)
- subd
  - BdNode, [26](#)
  - InternalBd, [125](#)
- swap
  - HeapSort, [100](#)
- swprint
  - DPMMainMPI, [53](#)
  - Interchange, [117](#)
- t1
  - DPMMainMPI, [53](#)
- t2
  - DPMMainMPI, [53](#)
- ta
  - EsquemaMEMPI, [77](#)
- tamanoBanda
  - MatrizDispersa, [152](#)
- time0
  - DPMMethod, [58](#)
- time1
  - DPMMethod, [58](#)
- time2
  - DPMMethod, [58](#)
- time3
  - DPMMethod, [58](#)
- toSubdomains
  - DualPrimal, [64](#)
  - Interchange, [115](#)
  - InterchangeMPI, [122](#)
- u
  - DPMMethod, [58](#)
- uf
  - LM1, [128](#)
  - LM2, [135](#)
  - PLM1, [181](#)
  - PLM2, [188](#)
  - PMF2, [201](#)
- ug
  - LM2, [135](#)
  - PLM2, [188](#)
- up
  - LM1, [128](#)

- LM2, [136](#)
- MF2, [162](#)
- PLM1, [181](#)
- PLM2, [189](#)
- PMF2, [201](#)
- v
  - CGM, [30](#)
  - DQGMRES, [62](#)
- VERTEX
  - Primal, [207](#)
  - RectSub, [219](#)
- val
  - ICGM, [104](#)
- vals
  - Properties, [213](#)
- var
  - Disc, [38](#)
  - Disc13, [41](#)
  - Disc14, [44](#)
  - Disc15, [47](#)
  - ExpVXY, [80](#)
  - ExpVXYZ, [83](#)
  - ExpX, [86](#)
  - ExpXY, [89](#)
  - ExpXYZ, [92](#)
  - fExpXY, [95](#)
  - NSfExpXY, [175](#)
  - NSfExpXYZ, [178](#)
  - SfExpXYZ, [223](#)
  - SinPiXSinPiYSinPiZ, [241](#)
  - SinPinxSinPiny, [226](#)
  - SinPinxSinPinySinPinz, [229](#)
  - SinPix, [232](#)
  - SinPixCosPiy, [235](#)
  - SinPixSinPiy, [238](#)
- VertEdgePrimal, [244](#)
  - isPrimal, [247](#)
  - name, [247](#)
  - VertEdgePrimal, [247](#)
- VertEdgePrimal.hpp, [323](#)
- VertPrimal, [247](#)
  - isPrimal, [250](#)
  - name, [250](#)
  - VertPrimal, [250](#)
- VertPrimal.hpp, [324](#)
- visualiza
  - MatrizDispersa, [152](#)
- visualizaMatricesInternas
  - MatrizDispersa, [152](#)
- X
  - RectSub, [220](#)
- x
  - RectSub, [220](#)
- XP
  - DualPrimal, [65](#)
- xnp
  - DPMainMPI, [53](#)
  - InterchangeMPI, [123](#)
- Y
  - RectSub, [220](#)
- YP
  - DualPrimal, [65](#)
- zero
  - DPMainMPI, [53](#)
  - Interchange, [117](#)