

LibHalfSpace

Generated by Doxygen 1.8.8

Sun Aug 21 2016 00:20:58

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	BE3D Class Reference	7
4.1.1	Constructor & Destructor Documentation	7
4.1.1.1	BE3D	7
4.1.1.2	BE3D	7
4.1.1.3	~BE3D	7
4.1.2	Member Function Documentation	8
4.1.2.1	DISPLACEMENT	8
4.1.2.2	DISPLACEMENT	8
4.1.2.3	GET_BC	8
4.1.2.4	GET_BV	8
4.1.2.5	GET_PARAM	8
4.1.2.6	NEW	8
4.1.2.7	PUT_BC	8
4.1.2.8	PUT_BV	8
4.1.2.9	SHIFT	8
4.1.2.10	STRAIN	8
4.1.2.11	STRESS	8
4.1.2.12	STRESS	8
4.2	BE3D_GEOM_PARAM Class Reference	8
4.2.1	Constructor & Destructor Documentation	9
4.2.1.1	BE3D_GEOM_PARAM	9
4.2.1.2	BE3D_GEOM_PARAM	9

4.2.2	Member Data Documentation	9
4.2.2.1	cc	9
4.2.2.2	delta_degree	9
4.2.2.3	flag	9
4.2.2.4	L	9
4.2.2.5	n	9
4.2.2.6	pc	9
4.2.2.7	phi_degree	9
4.2.2.8	td	9
4.2.2.9	ts	9
4.2.2.10	W	9
4.3	BE3D_PARAM Class Reference	9
4.3.1	Constructor & Destructor Documentation	10
4.3.1.1	BE3D_PARAM	10
4.3.1.2	~BE3D_PARAM	10
4.3.2	Member Data Documentation	10
4.3.2.1	GEOM	10
4.3.2.2	STRESS	10
4.4	BE3D_STRESS_PARAM Class Reference	10
4.4.1	Constructor & Destructor Documentation	10
4.4.1.1	BE3D_STRESS_PARAM	10
4.4.1.2	BE3D_STRESS_PARAM	10
4.5	CONSOLE Class Reference	10
4.5.1	Constructor & Destructor Documentation	11
4.5.1.1	CONSOLE	11
4.5.1.2	CONSOLE	11
4.5.1.3	CONSOLE	11
4.5.1.4	CONSOLE	11
4.5.1.5	CONSOLE	11
4.5.2	Member Function Documentation	11
4.5.2.1	GET_flag_print	11
4.5.3	Member Data Documentation	11
4.5.3.1	out_stream	12
4.6	DATAFILE Class Reference	12
4.6.1	Constructor & Destructor Documentation	12
4.6.1.1	DATAFILE	12
4.6.1.2	DATAFILE	12
4.6.1.3	DATAFILE	13
4.6.1.4	DATAFILE	13
4.6.1.5	DATAFILE	13

4.6.2	Member Function Documentation	13
4.6.2.1	GET_file_parameters	13
4.6.2.2	GET_file_parameters	13
4.6.2.3	GET_flag_datatype	13
4.7	DAVIS Class Reference	13
4.7.1	Constructor & Destructor Documentation	14
4.7.1.1	DAVIS	14
4.7.1.2	DAVIS	14
4.7.1.3	~DAVIS	14
4.7.2	Member Function Documentation	14
4.7.2.1	DISPLACEMENT	14
4.7.2.2	GET_pm	14
4.7.2.3	SET	14
4.8	discretization_param Struct Reference	15
4.8.1	Member Data Documentation	15
4.8.1.1	cc	15
4.8.1.2	flag	15
4.8.1.3	LV	15
4.8.1.4	N_1	15
4.8.1.5	N_2	15
4.8.1.6	N_3	15
4.8.1.7	n_matrix	15
4.8.1.8	NUM	15
4.8.1.9	NUM_COND	15
4.8.1.10	pc	15
4.8.1.11	TOTALE	15
4.8.1.12	vettore_delta	15
4.8.1.13	WV	15
4.9	FAULT Class Reference	16
4.9.1	Constructor & Destructor Documentation	16
4.9.1.1	FAULT	16
4.9.1.2	FAULT	16
4.9.1.3	~FAULT	16
4.9.2	Member Function Documentation	16
4.9.2.1	NEW	16
4.9.2.2	PRINT	17
4.9.2.3	PRINT	17
4.9.2.4	PRINT	17
4.9.2.5	PRINT_BV_COMP	17
4.9.2.6	PRINT_GEOM_PROP	17

4.9.2.7	PRINT_MAIN_FEATURES	17
4.9.2.8	PRINT_STRESS_PROP	17
4.9.3	Member Data Documentation	17
4.9.3.1	L	17
4.9.3.2	theta_degree	17
4.9.3.3	W	17
4.10	FIALKO Class Reference	17
4.10.1	Constructor & Destructor Documentation	18
4.10.1.1	FIALKO	18
4.10.1.2	FIALKO	18
4.10.1.3	~FIALKO	18
4.10.2	Member Function Documentation	18
4.10.2.1	DISPLACEMENT	18
4.10.2.2	DISPLACEMENT	18
4.10.2.3	PRINT	18
4.10.2.4	PRINT	19
4.10.2.5	SET	19
4.11	gravity_grid_param Struct Reference	19
4.11.1	Member Data Documentation	19
4.11.1.1	vett_TOT1	19
4.11.1.2	vett_TOT2	19
4.11.1.3	vett_TOT3	19
4.11.1.4	vett_TOT4	19
4.12	GRID Class Reference	19
4.12.1	Constructor & Destructor Documentation	20
4.12.1.1	GRID	20
4.12.1.2	GRID	20
4.12.1.3	~GRID	20
4.12.2	Member Data Documentation	20
4.12.2.1	N_1	20
4.12.2.2	N_2	20
4.12.2.3	N_3	20
4.12.2.4	NUM_COND	20
4.12.2.5	NX	20
4.12.2.6	NY	20
4.12.2.7	NZ	20
4.12.2.8	TOTAL	20
4.13	MEDIUM Class Reference	20
4.13.1	Constructor & Destructor Documentation	21
4.13.1.1	MEDIUM	21

4.13.1.2	MEDIUM	21
4.13.1.3	~MEDIUM	21
4.13.2	Member Function Documentation	21
4.13.2.1	GET_E	21
4.13.2.2	GET_K	21
4.13.2.3	GET_lambda	21
4.13.2.4	GET_mu	21
4.13.2.5	GET_nu	21
4.14	medium_param Struct Reference	21
4.14.1	Member Data Documentation	21
4.14.1.1	K	21
4.14.1.2	lambda	21
4.14.1.3	mu	21
4.14.1.4	nu	21
4.15	MOGI Class Reference	22
4.15.1	Constructor & Destructor Documentation	22
4.15.1.1	MOGI	22
4.15.1.2	MOGI	22
4.15.1.3	~MOGI	22
4.15.2	Member Function Documentation	22
4.15.2.1	DISPLACEMENT	22
4.15.2.2	DISPLACEMENT	22
4.15.2.3	PRINT	23
4.15.2.4	PRINT	23
4.15.2.5	SET	23
4.16	MOMENT Class Reference	23
4.16.1	Constructor & Destructor Documentation	24
4.16.1.1	MOMENT	24
4.16.1.2	MOMENT	24
4.16.1.3	~MOMENT	24
4.16.2	Member Function Documentation	24
4.16.2.1	DISPLACEMENT	24
4.16.2.2	PRINT	24
4.16.2.3	PRINT	24
4.16.2.4	SET	24
4.16.3	Member Data Documentation	24
4.16.3.1	M	24
4.17	obs_point_ti Struct Reference	24
4.17.1	Member Data Documentation	25
4.17.1.1	l1	25

4.17.1.2	n	25
4.17.1.3	ti	25
4.17.1.4	xn	25
4.17.1.5	yn	25
4.17.1.6	z	25
4.18	OKADA Class Reference	25
4.18.1	Detailed Description	26
4.18.2	Constructor & Destructor Documentation	26
4.18.2.1	OKADA	26
4.18.2.2	OKADA	26
4.18.2.3	~OKADA	26
4.18.3	Member Function Documentation	26
4.18.3.1	DISPLACEMENT	26
4.18.3.2	PRINT	26
4.18.3.3	PRINT	26
4.18.3.4	PRINT	26
4.18.3.5	SET	26
4.18.3.6	STRAIN	27
4.18.3.7	STRESS	27
4.19	OUTPUT Class Reference	27
4.19.1	Detailed Description	27
4.19.2	Member Function Documentation	27
4.19.2.1	GET_int_flag	27
4.19.3	Member Data Documentation	27
4.19.3.1	int_flag	28
4.20	PARAL Class Reference	28
4.20.1	Constructor & Destructor Documentation	28
4.20.1.1	PARAL	28
4.20.1.2	~PARAL	28
4.20.2	Member Function Documentation	28
4.20.2.1	PRINT	28
4.20.2.2	PRINT	28
4.20.2.3	PRINT	29
4.20.2.4	SET	29
4.20.2.5	SOLVE	29
4.21	PRESSURIZED Class Reference	29
4.21.1	Constructor & Destructor Documentation	30
4.21.1.1	PRESSURIZED	30
4.21.1.2	PRESSURIZED	30
4.21.1.3	~PRESSURIZED	30

4.21.2	Member Function Documentation	30
4.21.2.1	PRINT_MAIN_FEATURES	30
4.21.2.2	SET	30
4.21.3	Member Data Documentation	30
4.21.3.1	DP	30
4.22	SILL Class Reference	30
4.22.1	Constructor & Destructor Documentation	31
4.22.1.1	SILL	31
4.22.1.2	SILL	31
4.22.1.3	~SILL	31
4.22.2	Member Function Documentation	31
4.22.2.1	PRINT	31
4.22.2.2	PRINT	31
4.22.2.3	PRINT	31
4.23	SOURCE Class Reference	31
4.23.1	Constructor & Destructor Documentation	32
4.23.1.1	SOURCE	32
4.23.1.2	SOURCE	32
4.23.1.3	~SOURCE	32
4.23.2	Member Function Documentation	32
4.23.2.1	DISPLACEMENT	32
4.23.2.2	DISPLACEMENT	33
4.23.2.3	GET_label	33
4.23.2.4	GET_MEDIUM_PAR	33
4.23.2.5	GET_x0_y0_z0	33
4.23.2.6	PRINT	33
4.23.2.7	PRINT	33
4.23.2.8	PRINT	33
4.23.2.9	PUT_label	33
4.23.2.10	PUT_MEDIUM_PAR	33
4.23.2.11	PUT_x0_y0_z0	33
4.23.2.12	SET	34
4.23.2.13	STRAIN	34
4.23.2.14	STRESS	34
4.23.3	Member Data Documentation	34
4.23.3.1	label	34
4.23.3.2	MEDIUM_PAR	34
4.23.3.3	x0	34
4.23.3.4	y0	34
4.23.3.5	z0	34

4.24	SOURCE_3D Class Reference	34
4.24.1	Constructor & Destructor Documentation	35
4.24.1.1	SOURCE_3D	35
4.24.1.2	SOURCE_3D	35
4.24.1.3	~SOURCE_3D	35
4.25	SOURCE_BE3D Class Reference	35
4.25.1	Constructor & Destructor Documentation	36
4.25.1.1	SOURCE_BE3D	36
4.25.1.2	SOURCE_BE3D	36
4.25.1.3	~SOURCE_BE3D	36
4.25.2	Member Function Documentation	36
4.25.2.1	DISPLACEMENT	36
4.25.2.2	GET_FLAG_COMP	36
4.25.2.3	GET_NBE	36
4.25.2.4	GET_NDP	36
4.25.2.5	GET_phi	36
4.25.2.6	PRINT	36
4.25.2.7	PRINT	37
4.25.2.8	PRINT	37
4.25.2.9	PRINT_BV_COMP	37
4.25.2.10	PRINT_GEOM_PROP	37
4.25.2.11	PRINT_MAIN_FEATURES	37
4.25.2.12	PRINT_STRESS_PROP	37
4.25.2.13	PUT_x0_y0_z0	37
4.25.2.14	SET	37
4.25.2.15	SOLVE	37
4.25.2.16	STRAIN	38
4.25.2.17	STRESS	38
4.25.3	Member Data Documentation	38
4.25.3.1	BE	38
4.25.3.2	FLAG_COMP	38
4.25.3.3	GRID_PAR	38
4.25.3.4	NBE	38
4.25.3.5	NDP	38
4.25.3.6	phi_degree	38
4.26	source_param Struct Reference	38
4.26.1	Member Data Documentation	38
4.26.1.1	c1	38
4.26.1.2	Delta_P	39
4.26.1.3	h_1	39

4.26.1.4	h_2	39
4.26.1.5	h_3	39
4.26.1.6	max_side	39
4.26.1.7	type	39
4.27	struct_var Struct Reference	39
4.27.1	Member Data Documentation	40
4.27.1.1	j1A	40
4.27.1.2	J1A	40
4.27.1.3	j1B	40
4.27.1.4	j1C	40
4.27.1.5	j2A	40
4.27.1.6	J2A	40
4.27.1.7	j2B	40
4.27.1.8	j2C	40
4.27.1.9	j3A	40
4.27.1.10	J3A	40
4.27.1.11	j3B	40
4.27.1.12	j3C	41
4.27.1.13	k1A	41
4.27.1.14	K1A	41
4.27.1.15	k1B	41
4.27.1.16	k1C	41
4.27.1.17	k2A	41
4.27.1.18	K2A	41
4.27.1.19	k2B	41
4.27.1.20	k2C	41
4.27.1.21	k3A	41
4.27.1.22	K3A	41
4.27.1.23	k3B	41
4.27.1.24	k3C	41
4.27.1.25	l1A	41
4.27.1.26	L1A	41
4.27.1.27	l1B	41
4.27.1.28	l1C	41
4.27.1.29	l2A	41
4.27.1.30	L2A	41
4.27.1.31	l2B	41
4.27.1.32	l2C	41
4.27.1.33	l3A	41
4.27.1.34	L3A	41

4.27.1.35 l3B	41
4.27.1.36 l3C	41
4.27.1.37 u1A	41
4.27.1.38 U1A	41
4.27.1.39 u1B	41
4.27.1.40 u1C	42
4.27.1.41 U2A	42
4.27.1.42 u2A	42
4.27.1.43 u2B	42
4.27.1.44 u2C	42
4.27.1.45 U3A	42
4.27.1.46 u3A	42
4.27.1.47 u3B	42
4.27.1.48 u3C	42
4.27.1.49 Ux	42
4.27.1.50 Ux_x	42
4.27.1.51 Ux_y	42
4.27.1.52 Ux_z	42
4.27.1.53 Uy	42
4.27.1.54 Uy_x	42
4.27.1.55 Uy_y	42
4.27.1.56 Uy_z	42
4.27.1.57 Uz	42
4.27.1.58 Uz_x	42
4.27.1.59 Uz_y	42
4.27.1.60 Uz_z	42
4.28 subsource_param_tj Struct Reference	42
4.28.1 Member Data Documentation	43
4.28.1.1 c_cc	43
4.28.1.2 c_pc	43
4.28.1.3 delta_gradi	43
4.28.1.4 flag	43
4.28.1.5 l2	43
4.28.1.6 L	43
4.28.1.7 tj	43
4.28.1.8 W	43
4.28.1.9 x_cc	43
4.28.1.10 x_pc	43
4.28.1.11 y_cc	43
4.28.1.12 y_pc	43

4.29	YANG Class Reference	43
4.29.1	Constructor & Destructor Documentation	44
4.29.1.1	YANG	44
4.29.1.2	YANG	44
4.29.1.3	~YANG	44
4.29.2	Member Function Documentation	44
4.29.2.1	DISPLACEMENT	44
4.29.2.2	PRINT	44
4.29.2.3	PRINT	44
4.29.2.4	SET	45
5	File Documentation	47
5.1	SRC/BE3D.cpp File Reference	47
5.1.1	Function Documentation	49
5.1.1.1	DC3D	49
5.1.2	Variable Documentation	49
5.1.2.1	alpha	49
5.1.2.2	C	49
5.1.2.3	CDR	49
5.1.2.4	coef_alpha_1	49
5.1.2.5	coef_alpha_2	49
5.1.2.6	coefpi	49
5.1.2.7	COS_DELTA	49
5.1.2.8	COS_DELTA_2	49
5.1.2.9	CR	49
5.1.2.10	D	49
5.1.2.11	D2	49
5.1.2.12	D_11	49
5.1.2.13	E	49
5.1.2.14	Ep	49
5.1.2.15	EPS	49
5.1.2.16	Eta2	49
5.1.2.17	F	49
5.1.2.18	Fp	49
5.1.2.19	G	49
5.1.2.20	Gp	49
5.1.2.21	H	49
5.1.2.22	h	49
5.1.2.23	Hp	49
5.1.2.24	I_1	49

5.1.2.25	I_2	50
5.1.2.26	I_3	50
5.1.2.27	I_4	50
5.1.2.28	J_1	50
5.1.2.29	J_2	50
5.1.2.30	J_3	50
5.1.2.31	J_4	50
5.1.2.32	J_5	50
5.1.2.33	J_6	50
5.1.2.34	K_1	50
5.1.2.35	K_2	50
5.1.2.36	K_3	50
5.1.2.37	K_4	50
5.1.2.38	p	50
5.1.2.39	P	50
5.1.2.40	π	50
5.1.2.41	Pp	50
5.1.2.42	q	50
5.1.2.43	Q	50
5.1.2.44	q_2	50
5.1.2.45	q_3	50
5.1.2.46	Qp	50
5.1.2.47	qR	50
5.1.2.48	R	50
5.1.2.49	R_2	50
5.1.2.50	R_3	50
5.1.2.51	R_5	50
5.1.2.52	RpD	50
5.1.2.53	RpD_2	51
5.1.2.54	SIN_DELTA	51
5.1.2.55	SIN_DELTA_2	51
5.1.2.56	$Sing_iii$	51
5.1.2.57	$Sing_iv$	51
5.1.2.58	$Teta$	51
5.1.2.59	X	51
5.1.2.60	X_{11}	51
5.1.2.61	X_{32}	51
5.1.2.62	X_{53}	51
5.1.2.63	Ξ_2	51
5.1.2.64	Y	51

5.1.2.65	Y2	51
5.1.2.66	Y_0	51
5.1.2.67	Y_11	51
5.1.2.68	Y_32	51
5.1.2.69	Y_53	51
5.1.2.70	YY0	51
5.1.2.71	Z_0	51
5.1.2.72	Z_32	51
5.1.2.73	Z_53	51
5.2	SRC/BE3D.h File Reference	51
5.3	SRC/BE3D_GEOM_PARAM.h File Reference	52
5.4	SRC/BE3D_PARAM.h File Reference	52
5.5	SRC/BE3D_STRESS_PARAM.h File Reference	52
5.6	SRC/CONSOLE.cpp File Reference	52
5.7	SRC/CONSOLE.h File Reference	52
5.8	SRC/DATAFILE.cpp File Reference	53
5.9	SRC/DATAFILE.h File Reference	53
5.10	SRC/DAVIS.cpp File Reference	53
5.10.1	Function Documentation	53
5.10.1.1	A2M	53
5.10.2	Variable Documentation	53
5.10.2.1	pi	53
5.11	SRC/DAVIS.h File Reference	53
5.11.1	Function Documentation	54
5.11.1.1	A2M	54
5.11.1.2	rd	54
5.11.1.3	sgesv_	54
5.11.2	Variable Documentation	54
5.11.2.1	TABLE	54
5.11.2.2	TABLE_range	54
5.12	SRC/FAULT.cpp File Reference	54
5.13	SRC/FAULT.h File Reference	54
5.14	SRC/FIALKO.cpp File Reference	55
5.14.1	Function Documentation	55
5.14.1.1	fkernel	55
5.14.1.2	KERN	55
5.14.1.3	KG	55
5.14.1.4	Q	55
5.15	SRC/FIALKO.h File Reference	55
5.16	SRC/foo.h File Reference	55

5.16.1 Variable Documentation	56
5.16.1.1 max_dim	56
5.17 SRC/GRID.h File Reference	56
5.18 SRC/MEDIUM.cpp File Reference	56
5.19 SRC/MEDIUM.h File Reference	56
5.20 SRC/MOGL.cpp File Reference	56
5.21 SRC/MOGL.h File Reference	56
5.22 SRC/MOMENT.cpp File Reference	56
5.22.1 Function Documentation	57
5.22.1.1 G_hs	57
5.23 SRC/MOMENT.h File Reference	57
5.24 SRC/OKADA.cpp File Reference	57
5.25 SRC/OKADA.h File Reference	57
5.26 SRC/OUTPUT.h File Reference	57
5.27 SRC/PARAL.cpp File Reference	58
5.27.1 Function Documentation	58
5.27.1.1 dgesv_	58
5.27.1.2 F	58
5.27.1.3 f_BV_PROP	58
5.27.1.4 f_DISL_PROP	58
5.27.1.5 f_STRESS_PROP	58
5.27.1.6 PARAL_f_GEOM_PROP	58
5.28 SRC/PARAL.h File Reference	58
5.28.1 Function Documentation	59
5.28.1.1 dgesv_	59
5.29 SRC/PRESSURIZED.cpp File Reference	59
5.30 SRC/PRESSURIZED.h File Reference	59
5.31 SRC/PRINT_TEMPLATE.h File Reference	59
5.31.1 Function Documentation	60
5.31.1.1 print_displ_script	60
5.31.1.2 print_map_displ	60
5.31.1.3 print_map_radial_displ	60
5.31.1.4 print_map_strain	60
5.31.1.5 print_map_stress	60
5.31.1.6 print_maps	60
5.31.1.7 print_radial_displ_script	60
5.31.1.8 print_strain_script	60
5.31.1.9 print_stress_script	60
5.32 SRC/SILL.cpp File Reference	60
5.33 SRC/SILL.h File Reference	60

5.34 SRC/SOURCE.cpp File Reference	60
5.35 SRC/SOURCE.h File Reference	61
5.36 SRC/SOURCE_3D.cpp File Reference	61
5.37 SRC/SOURCE_3D.h File Reference	61
5.38 SRC/SOURCE_BE3D.cpp File Reference	61
5.38.1 Function Documentation	61
5.38.1.1 dgesv_	61
5.39 SRC/SOURCE_BE3D.h File Reference	62
5.40 SRC/Utilities.cpp File Reference	62
5.40.1 Function Documentation	62
5.40.1.1 Cycle_counter	62
5.40.1.2 DC3D	62
5.40.1.3 dc3d_	62
5.40.1.4 gsl_eigenvalueproblem	62
5.40.1.5 rd	62
5.40.1.6 ref_func	62
5.41 SRC/Utilities.h File Reference	63
5.41.1 Macro Definition Documentation	63
5.41.1.1 BIG	63
5.41.1.2 C1	63
5.41.1.3 C2	63
5.41.1.4 C3	63
5.41.1.5 C4	63
5.41.1.6 C5	63
5.41.1.7 C6	63
5.41.1.8 ERRTOL	63
5.41.1.9 FMAX	64
5.41.1.10 TINY	64
5.41.2 Function Documentation	64
5.41.2.1 Cycle_counter	64
5.41.2.2 DC3D	64
5.41.2.3 gsl_eigenvalueproblem	64
5.41.2.4 rd	64
5.41.2.5 ref_func	64
5.41.2.6 Rotazione_coord	64
5.41.2.7 Rotazione_coord	64
5.41.2.8 Rotazione_tensore	64
5.41.2.9 Rotazione_vettore	64
5.42 SRC/YANG.cpp File Reference	64
5.42.1 Function Documentation	64

5.42.1.1 sign	64
5.43 SRC/YANG.h File Reference	64

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

BE3D	7
BE3D_GEOM_PARAM	8
BE3D_PARAM	9
BE3D_STRESS_PARAM	10
discretization_param	15
gravity_grid_param	19
GRID	19
MEDIUM	20
medium_param	21
obs_point_ti	24
OUTPUT	27
CONSOLE	10
DATAFILE	12
SOURCE	31
SOURCE_3D	34
MOMENT	23
DAVIS	13
OKADA	25
PRESSURIZED	29
FIALKO	17
MOGI	22
YANG	43
SOURCE_BE3D	35
FAULT	16
PARAL	28
SILL	30
source_param	38
struct_var	39
subsource_param_tj	42

Chapter 2

Class Index

2.1 Class List

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

BE3D	7
BE3D_GEOM_PARAM	8
BE3D_PARAM	9
BE3D_STRESS_PARAM	10
CONSOLE	10
DATAFILE	12
DAVIS	13
discretization_param	15
FAULT	16
FIALKO	17
gravity_grid_param	19
GRID	19
MEDIUM	20
medium_param	21
MOGI	22
MOMENT	23
obs_point_ti	24
OKADA	
Brief description. Brief description continued	25
OUTPUT	
Brief description. Brief description continued	27
PARAL	28
PRESSURIZED	29
SILL	30
SOURCE	31
SOURCE_3D	34
SOURCE_BE3D	35
source_param	38
struct_var	39
subsource_param_tj	42
YANG	43

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

SRC/BE3D.cpp	47
SRC/BE3D.h	51
SRC/BE3D_GEOM_PARAM.h	52
SRC/BE3D_PARAM.h	52
SRC/BE3D_STRESS_PARAM.h	52
SRC/CONSOLE.cpp	52
SRC/CONSOLE.h	52
SRC/DATAFILE.cpp	53
SRC/DATAFILE.h	53
SRC/DAVIS.cpp	53
SRC/DAVIS.h	53
SRC/FAULT.cpp	54
SRC/FAULT.h	54
SRC/FIALKO.cpp	55
SRC/FIALKO.h	55
SRC/foo.h	55
SRC/GRID.h	56
SRC/MEDIUM.cpp	56
SRC/MEDIUM.h	56
SRC/MOGL.cpp	56
SRC/MOGL.h	56
SRC/MOMENT.cpp	56
SRC/MOMENT.h	57
SRC/OKADA.cpp	57
SRC/OKADA.h	57
SRC/OUTPUT.h	57
SRC/PARAL.cpp	58
SRC/PARAL.h	58
SRC/PRESSURIZED.cpp	59
SRC/PRESSURIZED.h	59
SRC/PRINT_TEMPLATE.h	59
SRC/SILL.cpp	60
SRC/SILL.h	60
SRC/SOURCE.cpp	60
SRC/SOURCE.h	61
SRC/SOURCE_3D.cpp	61
SRC/SOURCE_3D.h	61
SRC/SOURCE_BE3D.cpp	61

SRC/SOURCE_BE3D.h	62
SRC/Utilities.cpp	62
SRC/Utilities.h	63
SRC/YANG.cpp	64
SRC/YANG.h	64

Chapter 4

Class Documentation

4.1 BE3D Class Reference

```
#include <BE3D.h>
```

Public Member Functions

- **BE3D** ()
- **BE3D** (double xc, double yc, double zc, double xp, double yp, double zp, double BE_phi, double BE_delta, double BE_L, double BE_W, double n_x, double n_y, double n_z, double ts_x, double ts_y, double ts_z, double td_x, double td_y, double td_z)
- virtual **~BE3D** ()
- void **NEW** (double xc, double yc, double zc, double xp, double yp, double zp, double BE_phi, double BE_delta, double BE_L, double BE_W, double n_x, double n_y, double n_z, double ts_x, double ts_y, double ts_z, double td_x, double td_y, double td_z)
- void **DISPLACEMENT** (int FLAG_COMP, **MEDIUM** M_PAR, double xn, double yn, double z, double S_Displacement[], double D_Displacement[], double T_Displacement[])
- void **DISPLACEMENT** (int FLAG_COMP, **MEDIUM** M_PAR, double xn, double yn, double z, double Displacement[])
- void **STRESS** (int FLAG_COMP, **MEDIUM** M_PAR, double xn, double yn, double z, double S_Stress[], double D_Stress[], double T_Stress[])
- void **STRESS** (int FLAG_COMP, **MEDIUM** M_PAR, double xn, double yn, double z, double Stress[])
- void **STRAIN** (int FLAG_COMP, **MEDIUM** M_PAR, double xn, double yn, double z, double Strain[])
- void **GET_PARAM** (**BE3D_PARAM** &PARAM_o)
- void **GET_BV** (double &B1_o, double &B2_o, double &B3_o)
- void **PUT_BV** (double B1_i, double B2_i, double B3_i)
- void **GET_BC** (double &BC1_o, double &BC2_o, double &BC3_o)
- void **PUT_BC** (double BC1_i, double BC2_i, double BC3_i)
- void **SHIFT** (double Dx, double Dy, double Dz)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 **BE3D::BE3D** () [inline]

4.1.1.2 **BE3D::BE3D** (double xc, double yc, double zc, double xp, double yp, double zp, double BE_phi, double BE_delta, double BE_L, double BE_W, double n_x, double n_y, double n_z, double ts_x, double ts_y, double ts_z, double td_x, double td_y, double td_z) [inline]

4.1.1.3 virtual **BE3D::~BE3D** () [inline],[virtual]

4.1.2 Member Function Documentation

- 4.1.2.1 void BE3D::DISPLACEMENT (int *FLAG_COMP*, MEDIUM *M_PAR*, double *xn*, double *yn*, double *z*, double *S_Displacement*[], double *D_Displacement*[], double *T_Displacement*[])
- 4.1.2.2 void BE3D::DISPLACEMENT (int *FLAG_COMP*, MEDIUM *M_PAR*, double *xn*, double *yn*, double *z*, double *Displacement*[])
- 4.1.2.3 void BE3D::GET_BC (double & *BC1_o*, double & *BC2_o*, double & *BC3_o*) [inline]
- 4.1.2.4 void BE3D::GET_BV (double & *B1_o*, double & *B2_o*, double & *B3_o*) [inline]
- 4.1.2.5 void BE3D::GET_PARAM (BE3D_PARAM & *PARAM_o*) [inline]
- 4.1.2.6 void BE3D::NEW (double *xc*, double *yc*, double *zc*, double *xp*, double *yp*, double *zp*, double *BE_phi*, double *BE_delta*, double *BE_L*, double *BE_W*, double *n_x*, double *n_y*, double *n_z*, double *ts_x*, double *ts_y*, double *ts_z*, double *td_x*, double *td_y*, double *td_z*)
- 4.1.2.7 void BE3D::PUT_BC (double *BC1_i*, double *BC2_i*, double *BC3_i*) [inline]
- 4.1.2.8 void BE3D::PUT_BV (double *B1_i*, double *B2_i*, double *B3_i*) [inline]
- 4.1.2.9 void BE3D::SHIFT (double *Dx*, double *Dy*, double *Dz*)
- 4.1.2.10 void BE3D::STRAIN (int *FLAG_COMP*, MEDIUM *M_PAR*, double *xn*, double *yn*, double *z*, double *Strain*[])
- 4.1.2.11 void BE3D::STRESS (int *FLAG_COMP*, MEDIUM *M_PAR*, double *xn*, double *yn*, double *z*, double *S_Stress*[], double *D_Stress*[], double *T_Stress*[])
- 4.1.2.12 void BE3D::STRESS (int *FLAG_COMP*, MEDIUM *M_PAR*, double *xn*, double *yn*, double *z*, double *Stress*[])

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

- SRC/BE3D.h
- SRC/BE3D.cpp

4.2 BE3D_GEOM_PARAM Class Reference

```
#include <BE3D_GEOM_PARAM.h>
```

Public Member Functions

- [BE3D_GEOM_PARAM](#) ()
- [BE3D_GEOM_PARAM](#) (double *xc*, double *yc*, double *zc*, double *xp*, double *yp*, double *zp*, double *BE_phi*, double *BE_delta*, double *BE_L*, double *BE_W*, double *n_x*, double *n_y*, double *n_z*, double *ts_x*, double *ts_y*, double *ts_z*, double *td_x*, double *td_y*, double *td_z*)

Public Attributes

- double [cc](#) [3]
- double [pc](#) [3]
- double [n](#) [3]
- double [ts](#) [3]
- double [td](#) [3]

- double [phi_degree](#)
- double [delta_degree](#)
- double [L](#)
- double [W](#)
- int [flag](#)

4.2.1 Constructor & Destructor Documentation

4.2.1.1 `BE3D_GEOM_PARAM::BE3D_GEOM_PARAM ()` [\[inline\]](#)

4.2.1.2 `BE3D_GEOM_PARAM::BE3D_GEOM_PARAM (double xc, double yc, double zc, double xp, double yp, double zp, double BE_phi, double BE_delta, double BE_L, double BE_W, double n_x, double n_y, double n_z, double ts_x, double ts_y, double ts_z, double td_x, double td_y, double td_z)` [\[inline\]](#)

4.2.2 Member Data Documentation

4.2.2.1 `double BE3D_GEOM_PARAM::cc[3]`

4.2.2.2 `double BE3D_GEOM_PARAM::delta_degree`

4.2.2.3 `int BE3D_GEOM_PARAM::flag`

4.2.2.4 `double BE3D_GEOM_PARAM::L`

4.2.2.5 `double BE3D_GEOM_PARAM::n[3]`

4.2.2.6 `double BE3D_GEOM_PARAM::pc[3]`

4.2.2.7 `double BE3D_GEOM_PARAM::phi_degree`

4.2.2.8 `double BE3D_GEOM_PARAM::td[3]`

4.2.2.9 `double BE3D_GEOM_PARAM::ts[3]`

4.2.2.10 `double BE3D_GEOM_PARAM::W`

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

- [SRC/BE3D_GEOM_PARAM.h](#)

4.3 BE3D_PARAM Class Reference

```
#include <BE3D_PARAM.h>
```

Public Member Functions

- [BE3D_PARAM \(\)](#)
- virtual [~BE3D_PARAM \(\)](#)

Public Attributes

- [BE3D_GEOM_PARAM GEOM](#)
- [BE3D_STRESS_PARAM STRESS](#)

4.3.1 Constructor & Destructor Documentation

4.3.1.1 `BE3D_PARAM::BE3D_PARAM () [inline]`

4.3.1.2 `virtual BE3D_PARAM::~~BE3D_PARAM () [inline],[virtual]`

4.3.2 Member Data Documentation

4.3.2.1 `BE3D_GEOM_PARAM BE3D_PARAM::GEOM`

4.3.2.2 `BE3D_STRESS_PARAM BE3D_PARAM::STRESS`

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

- [SRC/BE3D_PARAM.h](#)

4.4 BE3D_STRESS_PARAM Class Reference

```
#include <BE3D_STRESS_PARAM.h>
```

Public Member Functions

- [BE3D_STRESS_PARAM](#) ()
- [BE3D_STRESS_PARAM](#) (double DP)

4.4.1 Constructor & Destructor Documentation

4.4.1.1 `BE3D_STRESS_PARAM::BE3D_STRESS_PARAM () [inline]`

4.4.1.2 `BE3D_STRESS_PARAM::BE3D_STRESS_PARAM (double DP) [inline]`

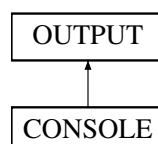
The documentation for this class was generated from the following file:

- [SRC/BE3D_STRESS_PARAM.h](#)

4.5 CONSOLE Class Reference

```
#include <CONSOLE.h>
```

Inheritance diagram for CONSOLE:



Public Member Functions

- [CONSOLE](#) ()
- [CONSOLE](#) (ostream &out_stream_i)

- [CONSOLE](#) (int int_flag_i)
- [CONSOLE](#) (int int_flag_i, ostream &out_stream_i)
- [CONSOLE](#) (int int_flag_i, ostream *out_stream_i, int flag_print=0)
- void [GET_flag_print](#) (int &flag_print_o)

Public Attributes

- ostream * [out_stream](#)

Additional Inherited Members

4.5.1 Constructor & Destructor Documentation

4.5.1.1 CONSOLE::CONSOLE ()

This method generates a [CONSOLE](#) object used to print to the standard output all the available information about the source

4.5.1.2 CONSOLE::CONSOLE (ostream & out_stream_i)

This method has the same output of [CONSOLE\(\)](#) but this output is redirected to the text file associated to out_stream.

4.5.1.3 CONSOLE::CONSOLE (int int_flag_i)

This method generates a [CONSOLE](#) object used to print to the standard output some selected information about the source:

0 - same output of [CONSOLE\(\)](#) method

1 - geometric properties

2 - stress properties

3 - Burgers vector components

10 - main features

According to the type of the source, some of the option are not available and in this case a warning message will be printed by the library.

4.5.1.4 CONSOLE::CONSOLE (int int_flag_i, ostream & out_stream_i)

This method has the same behavior of [CONSOLE\(int int_flag_i\)](#) but this output is redirected to the text file associated to out_stream.

4.5.1.5 CONSOLE::CONSOLE (int int_flag_i, ostream * out_stream_i, int flag_print = 0)

4.5.2 Member Function Documentation

4.5.2.1 void CONSOLE::GET_flag_print (int & flag_print_o) [inline]

4.5.3 Member Data Documentation

4.5.3.1 ostream* `CONSOLE::out_stream`

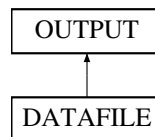
The documentation for this class was generated from the following files:

- SRC/[CONSOLE.h](#)
- SRC/[CONSOLE.cpp](#)

4.6 DATAFILE Class Reference

```
#include <DATAFILE.h>
```

Inheritance diagram for DATAFILE:



Public Member Functions

- [DATAFILE](#) (double *ri_i*, int *Nr_i*, double *rf_i*, string *extra*="")
- [DATAFILE](#) (string *extra*="")
- [DATAFILE](#) (int *int_flag_i*, string *extra*="")
- [DATAFILE](#) (double *DX1_i*, double *DX2_i*, int *N_1_i*, int *N_2_i*, string *extra*)
- [DATAFILE](#) (int *int_flag_i*, double *DX1_i*, double *DX2_i*, int *N_1_i*, int *N_2_i*, string *extra*)
- void [GET_flag_datatype](#) (int &*flag_datatype_o*)
- void [GET_file_parameters](#) (string &*filename_o*, int &*Nr_o*, double &*ri_o*, double &*rf_o*)
- void [GET_file_parameters](#) (string &*filename_o*, int &*N_1_o*, int &*N_2_o*, double &*DX1_o*, double &*DX2_o*)

Additional Inherited Members

4.6.1 Constructor & Destructor Documentation

4.6.1.1 `DATAFILE::DATAFILE (double ri_i, int Nr_i, double rf_i, string extra = " ")`

This method generates a [DATAFILE](#) object usable only by symmetric sources. When PRINT is invoked with this [DATAFILE](#) object as argument, then the radial and the vertical components of the displacement field at the free surface are computed for the selected source.

If PRINT is invoked by a asymmetric source, then a warning message is printed by the library

4.6.1.2 `DATAFILE::DATAFILE (string extra = " ")`

This method generates a [DATAFILE](#) object which builds a grid of observation points with these features:

- equal width on both x and y axes (*DX1* = 10 km, *DX2* = 10 km)
- equal density on both x and y axes (*N_1* = 100, *N_2* = 100)

When print is invoked with this [DATAFILE](#) object as argument, then displacement, strain and stress fields produced by the source are printed.

The optional string *extra* can be used to modify the default name of the files created by the PRINT method.

4.6.1.3 `DATAFILE::DATAFILE (int int_flag_i, string extra = " ")`

0 - all

1 - displacement

2 - strain

3 - stress

4.6.1.4 `DATAFILE::DATAFILE (double DX1_i, double DX2_i, int N_1_i, int N_2_i, string extra)`

This method is used to build `DATAFILE` object which defines a non standard grid of observation points.

Displacement, strain and stress fields are selected to be printed and in this case the string must be specified in order to distinguish the non standard grid.

4.6.1.5 `DATAFILE::DATAFILE (int int_flag_i, double DX1_i, double DX2_i, int N_1_i, int N_2_i, string extra)`

Same behavior of the previous method but `int_flag` allows to select the desired output:

0 - all

1 - displacement

2 - strain

3 - stress

4.6.2 Member Function Documentation

4.6.2.1 `void DATAFILE::GET_file_parameters (string & filename_o, int & Nr_o, double & ri_o, double & rf_o)`

4.6.2.2 `void DATAFILE::GET_file_parameters (string & filename_o, int & N_1_o, int & N_2_o, double & DX1_o, double & DX2_o)`

4.6.2.3 `void DATAFILE::GET_flag_datatype (int & flag_datatype_o)`

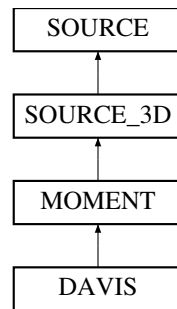
The documentation for this class was generated from the following files:

- [SRC/DATAFILE.h](#)
- [SRC/DATAFILE.cpp](#)

4.7 DAVIS Class Reference

```
#include <DAVIS.h>
```

Inheritance diagram for DAVIS:



Public Member Functions

- [DAVIS](#) ()
- [DAVIS](#) (string *label*, double *x0_i*, double *y0_i*, double *z0_i*)
- virtual [~DAVIS](#) ()
- void [SET](#) ([MEDIUM](#) *medium_par_i*, double *MXX*, double *MYY*, double *MZZ*, double *MXY*, double *MXZ*, double *MYZ*)
- void [DISPLACEMENT](#) (double *x*, double *y*, double *z*, double *U[]*)
- void [GET_pm](#) (double &*M1*, double &*M2*, double &*M3*)

Additional Inherited Members

4.7.1 Constructor & Destructor Documentation

4.7.1.1 [DAVIS::DAVIS](#) ()

4.7.1.2 [DAVIS::DAVIS](#) (string *label*, double *x0_i*, double *y0_i*, double *z0_i*) [inline]

4.7.1.3 [DAVIS::~~DAVIS](#) () [virtual]

4.7.2 Member Function Documentation

4.7.2.1 void [DAVIS::DISPLACEMENT](#) (double *x*, double *y*, double *z*, double *U[]*) [inline],[virtual]

Computation of the displacement components ($U_x=U[0]$, $U_y=U[1]$, $U_z=U[2]$) at the observation point (*x,y,z*)

Reimplemented from [SOURCE](#).

4.7.2.2 void [DAVIS::GET_pm](#) (double &*M1*, double &*M2*, double &*M3*) [inline]

This method is used to retrieve the principal moments of the moment tensor.

4.7.2.3 void [DAVIS::SET](#) ([MEDIUM](#) *medium_par_i*, double *MXX*, double *MYY*, double *MZZ*, double *MXY*, double *MXZ*, double *MYZ*)

The medium parameters are assigned to the source. We also assigns the moment tensor associated to the source. This method checks if the moment tensor is compatible with a [DAVIS](#) source.

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

- [SRC/DAVIS.h](#)
- [SRC/DAVIS.cpp](#)

4.8 discretization_param Struct Reference

```
#include <foo.h>
```

Public Attributes

- int [NUM](#)
- double [cc](#) [[max_dim](#)][3]
- double [pc](#) [[max_dim](#)][3]
- double [n_matrix](#) [[max_dim](#)][3]
- double [vettore_delta](#) [[max_dim](#)]
- double [LV](#) [[max_dim](#)]
- double [WV](#) [[max_dim](#)]
- int [flag](#) [[max_dim](#)]
- int [N_1](#)
- int [N_2](#)
- int [N_3](#)
- int [TOTALE](#)
- int [NUM_COND](#)

4.8.1 Member Data Documentation

4.8.1.1 double discretization_param::cc[max_dim][3]

4.8.1.2 int discretization_param::flag[max_dim]

4.8.1.3 double discretization_param::LV[max_dim]

4.8.1.4 int discretization_param::N_1

4.8.1.5 int discretization_param::N_2

4.8.1.6 int discretization_param::N_3

4.8.1.7 double discretization_param::n_matrix[max_dim][3]

4.8.1.8 int discretization_param::NUM

4.8.1.9 int discretization_param::NUM_COND

4.8.1.10 double discretization_param::pc[max_dim][3]

4.8.1.11 int discretization_param::TOTALE

4.8.1.12 double discretization_param::vettore_delta[max_dim]

4.8.1.13 double discretization_param::WV[max_dim]

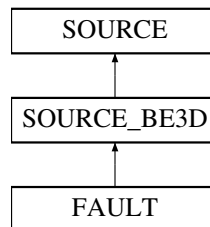
The documentation for this struct was generated from the following file:

- SRC/[foo.h](#)

4.9 FAULT Class Reference

```
#include <FAULT.h>
```

Inheritance diagram for FAULT:



Public Member Functions

- [FAULT](#) ()
- [FAULT](#) (string *s_label*, double *x0*, double *y0*, double *z0*, double *L*, double *W*, double *phi_degree*, double *theta_degree*, int *N*)
- virtual [~FAULT](#) ()
- void [PRINT](#) ()
- void [PRINT](#) ([CONSOLE](#) &out)
- void [PRINT](#) ([DATAFILE](#) &out)

Protected Member Functions

- void [NEW](#) (double *L*, double *W*, double *delta_gradi*)
- void [PRINT_GEOM_PROP](#) (ostream *out_stream)
- void [PRINT_STRESS_PROP](#) (ostream *stream)
- void [PRINT_BV_COMP](#) (ostream *stream)
- void [PRINT_MAIN_FEATURES](#) (ostream *out_stream, int flag_print=0)

Protected Attributes

- double [L](#)
- double [W](#)
- double [theta_degree](#)

4.9.1 Constructor & Destructor Documentation

4.9.1.1 [FAULT::FAULT](#) ()

4.9.1.2 [FAULT::FAULT](#) (string *s_label*, double *x0*, double *y0*, double *z0*, double *L*, double *W*, double *phi_degree*, double *theta_degree*, int *N*) `[inline]`

4.9.1.3 [FAULT::~~FAULT](#) () `[virtual]`

4.9.2 Member Function Documentation

4.9.2.1 void [FAULT::NEW](#) (double *L*, double *W*, double *delta_gradi*) `[protected]`

4.9.2.2 void FAULT::PRINT () [inline],[virtual]

Generation of a datafile and a gnuplot script suitable to obtain a graphic representation of the source model geometry

Reimplemented from [SOURCE_BE3D](#).

4.9.2.3 void FAULT::PRINT (**CONSOLE** & *out*) [virtual]

Print to standard output useful information about the source

Reimplemented from [SOURCE_BE3D](#).

4.9.2.4 void FAULT::PRINT (**DATAFILE** & *out*) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Reimplemented from [SOURCE_BE3D](#).

4.9.2.5 void FAULT::PRINT_BV_COMP (ostream * *stream*) [protected]

4.9.2.6 void FAULT::PRINT_GEOM_PROP (ostream * *out_stream*) [protected]

4.9.2.7 void FAULT::PRINT_MAIN_FEATURES (ostream * *out_stream*, int *flag_print* = 0) [protected]

4.9.2.8 void FAULT::PRINT_STRESS_PROP (ostream * *stream*) [protected]

4.9.3 Member Data Documentation

4.9.3.1 double FAULT::L [protected]

4.9.3.2 double FAULT::theta_degree [protected]

4.9.3.3 double FAULT::W [protected]

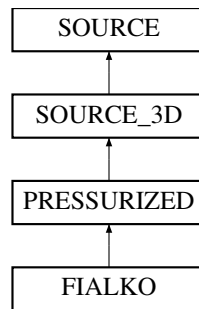
The documentation for this class was generated from the following files:

- [SRC/FAULT.h](#)
- [SRC/FAULT.cpp](#)

4.10 FIALKO Class Reference

```
#include <FIALKO.h>
```

Inheritance diagram for FIALKO:



Public Member Functions

- [FIALKO](#) ()
- [FIALKO](#) (string *label_i*, double *x0_i*, double *y0_i*, double *z0_i*, double *R_i*)
- virtual [~FIALKO](#) ()
- void [SET](#) ([MEDIUM](#) *MEDIUM_PAR_i*, double *DP_i*)
- void [DISPLACEMENT](#) (double *r*, double *z*, double &*Ur*, double &*Uz*)
- void [DISPLACEMENT](#) (double *x*, double *y*, double *z*, double *U[]*)
- void [PRINT](#) ([CONSOLE](#) &*out*)
- void [PRINT](#) ([DATAFILE](#) &*out*)

Additional Inherited Members

4.10.1 Constructor & Destructor Documentation

4.10.1.1 [FIALKO::FIALKO](#) ()

4.10.1.2 [FIALKO::FIALKO](#) (string *label_i*, double *x0_i*, double *y0_i*, double *z0_i*, double *R_i*) [inline]

4.10.1.3 [FIALKO::~~FIALKO](#) () [virtual]

Association of the medium param with the source buried in the Half-space < We also fix the assigned overpressure (*DP_i*).

4.10.2 Member Function Documentation

4.10.2.1 void [FIALKO::DISPLACEMENT](#) (double *r*, double *z*, double &*Ur*, double &*Uz*) [virtual]

Computation of the displacement components (*Ux*=*U*[0], *Uy*=*U*[1], *Uz*=*U*[2]) at the observation point (*x,y,z*)

Reimplemented from [SOURCE](#).

4.10.2.2 void [FIALKO::DISPLACEMENT](#) (double *x*, double *y*, double *z*, double *U[]*) [virtual]

Print to standard output useful information about the source

Reimplemented from [SOURCE](#).

4.10.2.3 void [FIALKO::PRINT](#) ([CONSOLE](#) & *out*) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Reimplemented from [SOURCE](#).

4.10.2.4 void FIALKO::PRINT (DATAFILE & out) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Implements [SOURCE](#).

4.10.2.5 void FIALKO::SET (MEDIUM MEDIUM_PAR_i, double DP_i) [virtual]

Computation of the displacement components (Ur, radial component and Uz, vertical component) at the observation point of polar coordinates (r,z)

Implements [PRESSURIZED](#).

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

- [SRC/FIALKO.h](#)
- [SRC/FIALKO.cpp](#)

4.11 gravity_grid_param Struct Reference

```
#include <foo.h>
```

Public Attributes

- double [vett_TOT1](#) [4]
- double [vett_TOT2](#) [4]
- double [vett_TOT3](#) [4]
- double [vett_TOT4](#) [4]

4.11.1 Member Data Documentation

4.11.1.1 double gravity_grid_param::vett_TOT1[4]

4.11.1.2 double gravity_grid_param::vett_TOT2[4]

4.11.1.3 double gravity_grid_param::vett_TOT3[4]

4.11.1.4 double gravity_grid_param::vett_TOT4[4]

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

- [SRC/foo.h](#)

4.12 GRID Class Reference

```
#include <GRID.h>
```

Public Member Functions

- [GRID](#) ()
- [GRID](#) (int N_1_i, int N_2_i, int N_3_i, int N_COND=0)
- [~GRID](#) ()

Public Attributes

- int [N_1](#)
- int [N_2](#)
- int [N_3](#)
- int [NX](#)
- int [NY](#)
- int [NZ](#)
- int [TOTAL](#)
- int [NUM_COND](#)

4.12.1 Constructor & Destructor Documentation

4.12.1.1 `GRID::GRID ()` [\[inline\]](#)

4.12.1.2 `GRID::GRID (int N_1_i, int N_2_i, int N_3_i, int N_COND = 0)` [\[inline\]](#)

4.12.1.3 `GRID::~~GRID ()` [\[inline\]](#)

4.12.2 Member Data Documentation

4.12.2.1 `int GRID::N_1`

4.12.2.2 `int GRID::N_2`

4.12.2.3 `int GRID::N_3`

4.12.2.4 `int GRID::NUM_COND`

4.12.2.5 `int GRID::NX`

4.12.2.6 `int GRID::NY`

4.12.2.7 `int GRID::NZ`

4.12.2.8 `int GRID::TOTAL`

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

- [SRC/GRID.h](#)

4.13 MEDIUM Class Reference

```
#include <MEDIUM.h>
```

Public Member Functions

- [MEDIUM](#) ()
- [MEDIUM](#) (double mu, double nu)
- virtual [~MEDIUM](#) ()
- double [GET_mu](#) ()
- double [GET_nu](#) ()
- double [GET_lambda](#) ()

- double [GET_K\(\)](#)
- double [GET_E\(\)](#)

4.13.1 Constructor & Destructor Documentation

4.13.1.1 `MEDIUM::MEDIUM ()` `[inline]`

4.13.1.2 `MEDIUM::MEDIUM (double mu, double nu)`

4.13.1.3 `virtual MEDIUM::~~MEDIUM ()` `[inline]`, `[virtual]`

4.13.2 Member Function Documentation

4.13.2.1 `double MEDIUM::GET_E ()`

4.13.2.2 `double MEDIUM::GET_K ()`

4.13.2.3 `double MEDIUM::GET_lambda ()`

4.13.2.4 `double MEDIUM::GET_mu ()`

4.13.2.5 `double MEDIUM::GET_nu ()`

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

- [SRC/MEDIUM.h](#)
- [SRC/MEDIUM.cpp](#)

4.14 medium_param Struct Reference

```
#include <foo.h>
```

Public Attributes

- double [mu](#)
- double [lambda](#)
- double [nu](#)
- double [K](#)

4.14.1 Member Data Documentation

4.14.1.1 `double medium_param::K`

4.14.1.2 `double medium_param::lambda`

4.14.1.3 `double medium_param::mu`

4.14.1.4 `double medium_param::nu`

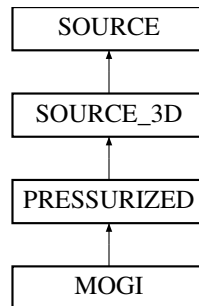
The documentation for this struct was generated from the following file:

- [SRC/foo.h](#)

4.15 MOGI Class Reference

```
#include <MOGI.h>
```

Inheritance diagram for MOGI:



Public Member Functions

- [MOGI](#) ()
- [MOGI](#) (string label_i, double x0_i, double y0_i, double z0_i, double R_i)
- virtual [~MOGI](#) ()
- void [SET](#) ([MEDIUM](#) MEDIUM_PAR_i, double P_i)
- void [DISPLACEMENT](#) (double r, double z, double &Ur, double &Uz)
- void [DISPLACEMENT](#) (double x, double y, double z, double U[])
- void [PRINT](#) ([CONSOLE](#) &out)
- void [PRINT](#) ([DATAFILE](#) &out)

Additional Inherited Members

4.15.1 Constructor & Destructor Documentation

4.15.1.1 [MOGI::MOGI](#) ()

4.15.1.2 [MOGI::MOGI](#) (string label_i, double x0_i, double y0_i, double z0_i, double R_i) [\[inline\]](#)

4.15.1.3 [MOGI::~~MOGI](#) () [\[virtual\]](#)

4.15.2 Member Function Documentation

4.15.2.1 void [MOGI::DISPLACEMENT](#) (double r, double z, double &Ur, double &Uz) [\[virtual\]](#)

Computation of the displacement components (Ur radial component, Uz vertical component) at the observation point (x,y,z) (available only for axially symmetric sources)

Reimplemented from [SOURCE](#).

4.15.2.2 void [MOGI::DISPLACEMENT](#) (double x, double y, double z, double U[]) [\[virtual\]](#)

Computation of the displacement components (Ux=U[0], Uy=U[1], Uz=U[2]) at the observation point (x,y,z)

Reimplemented from [SOURCE](#).

4.15.2.3 void MOGI::PRINT (**CONSOLE** & *out*) [virtual]

Print to standard output useful information about the source

Reimplemented from [SOURCE](#).

4.15.2.4 void MOGI::PRINT (**DATAFILE** & *out*) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Implements [SOURCE](#).

4.15.2.5 void MOGI::SET (**MEDIUM** *MEDIUM_PAR_i*, double *P_i*) [virtual]

Implements [PRESSURIZED](#).

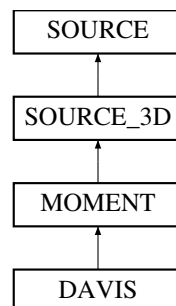
The documentation for this class was generated from the following files:

- [SRC/MOGI.h](#)
- [SRC/MOGI.cpp](#)

4.16 MOMENT Class Reference

```
#include <MOMENT.h>
```

Inheritance diagram for MOMENT:



Public Member Functions

- [MOMENT](#) ()
- [MOMENT](#) (string *label*, double *x0_i*, double *y0_i*, double *z0_i*)
- virtual [~MOMENT](#) ()
- void [SET](#) ([MEDIUM](#) *medium_par_i*, double *MXX*, double *MYX*, double *MZZ*, double *MXZ*, double *MYZ*)
- void [DISPLACEMENT](#) (double *x*, double *y*, double *z*, double *U*[])
- void [PRINT](#) ([CONSOLE](#) &*out*)
- void [PRINT](#) ([DATAFILE](#) &*out*)

Protected Attributes

- double [M](#) [6]

4.16.1 Constructor & Destructor Documentation

4.16.1.1 `MOMENT::MOMENT ()`

4.16.1.2 `MOMENT::MOMENT (string label, double x0_i, double y0_i, double z0_i)` `[inline]`

4.16.1.3 `MOMENT::~~MOMENT ()` `[virtual]`

4.16.2 Member Function Documentation

4.16.2.1 `void MOMENT::DISPLACEMENT (double x, double y, double z, double U[])` `[virtual]`

Computation of the displacement components ($U_x=U[0]$, $U_y=U[1]$, $U_z=U[2]$) at the observation point (x,y,z)

Reimplemented from [SOURCE](#).

4.16.2.2 `void MOMENT::PRINT (CONSOLE & out)` `[virtual]`

Print to standard output useful information about the source

Reimplemented from [SOURCE](#).

4.16.2.3 `void MOMENT::PRINT (DATAFILE & out)` `[virtual]`

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Implements [SOURCE](#).

4.16.2.4 `void MOMENT::SET (MEDIUM medium_par_i, double MXX, double MYY, double MZZ, double MXY, double MXZ, double MYZ)` `[inline]`

The medium parameters are assigned to the source. We also assigns the moment tensor associated to the source.

4.16.3 Member Data Documentation

4.16.3.1 `double MOMENT::M[6]` `[protected]`

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

- [SRC/MOMENT.h](#)
- [SRC/MOMENT.cpp](#)

4.17 obs_point_ti Struct Reference

```
#include <foo.h>
```

Public Attributes

- double [xn](#)
- double [yn](#)
- double [z](#)
- double [n](#) [3]

- int [ti](#)
- int [l1](#)

4.17.1 Member Data Documentation

4.17.1.1 int `obs_point_ti::l1`

4.17.1.2 double `obs_point_ti::n[3]`

4.17.1.3 int `obs_point_ti::ti`

4.17.1.4 double `obs_point_ti::xn`

4.17.1.5 double `obs_point_ti::yn`

4.17.1.6 double `obs_point_ti::z`

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

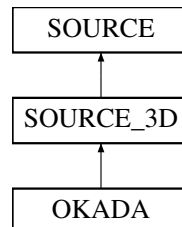
- SRC/[foo.h](#)

4.18 OKADA Class Reference

Brief description. Brief description continued.

```
#include <OKADA.h>
```

Inheritance diagram for OKADA:



Public Member Functions

- [OKADA](#) ()
- [OKADA](#) (string [label](#), double `x0_i`, double `y0_i`, double `z0_i`, double `L`, double `W`, double `theta_deg`, double `phi_deg`)
- virtual [~OKADA](#) ()
- void [SET](#) ([MEDIUM](#) `MEDIUM_PAR_i`, double `U_s`, double `U_d`, double `U_t`, int `FLAG_COMP=2`)
- void [DISPLACEMENT](#) (double `x`, double `y`, double `z`, double `U[]`)
- void [STRAIN](#) (double `x`, double `y`, double `z`, double `Strain[]`)
- void [STRESS](#) (double `x`, double `y`, double `z`, double `Stress[]`)
- void [PRINT](#) ()
- void [PRINT](#) ([CONSOLE](#) &out)
- void [PRINT](#) ([DATAFILE](#) &out)

Additional Inherited Members

4.18.1 Detailed Description

Brief description. Brief description continued.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 OKADA::OKADA ()

4.18.2.2 OKADA::OKADA (string *label*, double *x0_i*, double *y0_i*, double *z0_i*, double *L*, double *W*, double *theta_deg*, double *phi_deg*) [inline]

4.18.2.3 OKADA::~~OKADA () [virtual]

4.18.3 Member Function Documentation

4.18.3.1 void OKADA::DISPLACEMENT (double *x*, double *y*, double *z*, double *U[]*) [virtual]

Computation of the displacement components ($U_x=U[0]$, $U_y=U[1]$, $U_z=U[2]$) at the observation point (x,y,z)

Reimplemented from [SOURCE](#).

4.18.3.2 void OKADA::PRINT () [virtual]

Generation of a datafile and a gnuplot script suitable to obtain a graphic representation of the source model geometry

Reimplemented from [SOURCE](#).

4.18.3.3 void OKADA::PRINT (**CONSOLE** & *out*) [virtual]

Print to standard output useful information about the source

Reimplemented from [SOURCE](#).

4.18.3.4 void OKADA::PRINT (**DATAFILE** & *out*) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Implements [SOURCE](#).

4.18.3.5 void OKADA::SET (**MEDIUM** *MEDIUM_PAR_i*, double *U_s*, double *U_d*, double *U_t*, int *FLAG_COMP* = 2)

Association of the medium param with the source buried in the Half-space.

We also fix:

- the strike slip component (U_s)
- the dip slip component (U_d)
- the tensile component (U_t) The value assigned to `FLAG_COMP` assigns the Okada implementation adopted:
- 1, Ferrari et Al. implementation
- 2, Okada implementation (dc3d.f). If the value is not assigned, the default value is 2.

4.18.3.6 `void OKADA::STRAIN (double x, double y, double z, double Strain[]) [virtual]`

Computation of the strain tensor components ($S_{xx}=Strain[0]$, $S_{yy}=Strain[1]$, $S_{zz}=Strain[2]$, $S_{xy}=Strain[3]$, $S_{xz}=Strain[4]$, $S_{yz}=Strain[5]$) at the observation point (x,y,z)

Reimplemented from [SOURCE](#).

4.18.3.7 `void OKADA::STRESS (double x, double y, double z, double Stress[]) [virtual]`

Computation of the stress tensor components ($S_{xx}=Stress[0]$, $S_{yy}=Stress[1]$, $S_{zz}=Stress[2]$, $S_{xy}=Stress[3]$, $S_{xz}=Stress[4]$, $S_{yz}=Stress[5]$) at the observation point (x,y,z)

Reimplemented from [SOURCE](#).

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

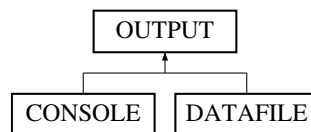
- [SRC/OKADA.h](#)
- [SRC/OKADA.cpp](#)

4.19 OUTPUT Class Reference

Brief description. Brief description continued.

```
#include <OUTPUT.h>
```

Inheritance diagram for OUTPUT:



Public Member Functions

- `void GET_int_flag (int &int_flag_o)`

Protected Attributes

- `int int_flag`

4.19.1 Detailed Description

Brief description. Brief description continued.

Detailed description starts here.

4.19.2 Member Function Documentation

4.19.2.1 `void OUTPUT::GET_int_flag (int &int_flag_o) [inline]`

4.19.3 Member Data Documentation

4.19.3.1 int OUTPUT::int_flag [protected]

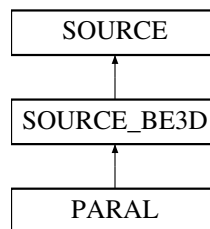
The documentation for this class was generated from the following file:

- SRC/OUTPUT.h

4.20 PARAL Class Reference

```
#include <PARAL.h>
```

Inheritance diagram for PARAL:



Public Member Functions

- [PARAL](#) (string *s_label*, double *x0*, double *y0*, double *z0*, double *h1*, double *h2*, double *h3*, double *fi*, int NUM)
- virtual [~PARAL](#) ()
- void [SET](#) ([MEDIUM](#) MEDIUM_PAR_i, int option_flag, double INPUT_1, double INPUT_2, double INPUT_3, int FLAG_COMP_i=1)
- void [SOLVE](#) ()
- void [PRINT](#) ()
- void [PRINT](#) ([CONSOLE](#) &out)
- void [PRINT](#) ([DATAFILE](#) &out)

Additional Inherited Members

4.20.1 Constructor & Destructor Documentation

4.20.1.1 **PARAL::PARAL** (string *s_label*, double *x0*, double *y0*, double *z0*, double *h1*, double *h2*, double *h3*, double *fi*, int *NUM*) [inline]

4.20.1.2 **virtual PARAL::~~PARAL** () [inline],[virtual]

4.20.2 Member Function Documentation

4.20.2.1 **void PARAL::PRINT** () [inline],[virtual]

Generation of a datafile and a gnuplot script suitable to obtain a graphic representation of the source model geometry

Reimplemented from [SOURCE_BE3D](#).

4.20.2.2 **void PARAL::PRINT** ([CONSOLE](#) & *out*) [virtual]

Print to standard output useful information about the source

Reimplemented from [SOURCE_BE3D](#).

4.20.2.3 void PARAL::PRINT (DATAFILE & out) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Reimplemented from [SOURCE_BE3D](#).

4.20.2.4 void PARAL::SET (MEDIUM MEDIUM_PAR_i, int option_flag, double INPUT_1, double INPUT_2, double INPUT_3, int FLAG_COMP_i = 1) [virtual]

Association of the medium param with the source buried in the Half-space This method uses the integer variable option_flag to choose the type of input expected:

- option_flag = 1, we consider the direct problem in fracture theory; a uniform slip is assigned over the source boundaries, with strike-slip component equal to INPUT_1, with dip-slip component equal to INPUT_2 and with tensile component with strike-slip component equal to INPUT_3;
- option_flag = 2, we consider the inverse problem in fracture theory, uniform boundary conditions are assigned over source boundaries. The optional variable FLAG_COMP_i can be specified in order to select the Okada implementation adopted:
- FLAG_COMP_i = 1, the implementation given by Ferrari et Al. (default option);
- FLAG_COMP_i = 2, the original implementation by Okada (dc3d.f code)

Reimplemented from [SOURCE_BE3D](#).

4.20.2.5 void PARAL::SOLVE () [virtual]

SOLVE implements the discontinuity element method (BEM). The implementation of the method for the [PARAL](#) model source is designed to solve an issue related to closed sources.

Reimplemented from [SOURCE_BE3D](#).

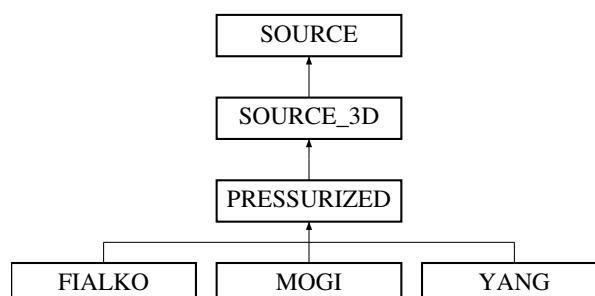
The documentation for this class was generated from the following files:

- SRC/[PARAL.h](#)
- SRC/[PARAL.cpp](#)

4.21 PRESSURIZED Class Reference

```
#include <PRESSURIZED.h>
```

Inheritance diagram for PRESSURIZED:



Public Member Functions

- [PRESSURIZED](#) ()
- [PRESSURIZED](#) (string *s_label*, double *x0*, double *y0*, double *z0*)
- virtual [~PRESSURIZED](#) ()
- virtual void [SET](#) ([MEDIUM](#) *MEDIUM_PAR_i*, double *P_i*)=0

Protected Member Functions

- void [PRINT_MAIN_FEATURES](#) (ostream **out_stream*, int *flag_print*=0)

Protected Attributes

- double [DP](#)

4.21.1 Constructor & Destructor Documentation

4.21.1.1 [PRESSURIZED::PRESSURIZED](#) ()

4.21.1.2 [PRESSURIZED::PRESSURIZED](#) (string *s_label*, double *x0*, double *y0*, double *z0*) [inline]

4.21.1.3 [PRESSURIZED::~~PRESSURIZED](#) () [virtual]

4.21.2 Member Function Documentation

4.21.2.1 void [PRESSURIZED::PRINT_MAIN_FEATURES](#) (ostream * *out_stream*, int *flag_print* = 0) [protected]

4.21.2.2 virtual void [PRESSURIZED::SET](#) ([MEDIUM](#) *MEDIUM_PAR_i*, double *P_i*) [pure virtual]

Implemented in [YANG](#), [FIALKO](#), and [MOGI](#).

4.21.3 Member Data Documentation

4.21.3.1 double [PRESSURIZED::DP](#) [protected]

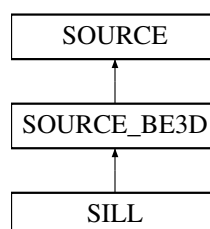
The documentation for this class was generated from the following files:

- SRC/[PRESSURIZED.h](#)
- SRC/[PRESSURIZED.cpp](#)

4.22 SILL Class Reference

```
#include <SILL.h>
```

Inheritance diagram for SILL:



Public Member Functions

- [SILL](#) ()
- [SILL](#) (string *s_label*, double *x0*, double *y0*, double *z0*, double *a*, double *b*, double *phi_degree*, double *theta_degree*, int *N*)
- virtual [~SILL](#) ()
- void [PRINT](#) ()
- void [PRINT](#) ([CONSOLE](#) &out)
- void [PRINT](#) ([DATAFILE](#) &out)

Additional Inherited Members

4.22.1 Constructor & Destructor Documentation

4.22.1.1 [SILL::SILL](#) ()

4.22.1.2 [SILL::SILL](#) (string *s_label*, double *x0*, double *y0*, double *z0*, double *a*, double *b*, double *phi_degree*, double *theta_degree*, int *N*) [inline]

4.22.1.3 [SILL::~~SILL](#) () [virtual]

4.22.2 Member Function Documentation

4.22.2.1 void [SILL::PRINT](#) () [inline],[virtual]

Generation of a datafile and a gnuplot script suitable to obtain a graphic representation of the source model geometry

Reimplemented from [SOURCE_BE3D](#).

4.22.2.2 void [SILL::PRINT](#) ([CONSOLE](#) & *out*) [virtual]

Print to standard output useful information about the source

Reimplemented from [SOURCE_BE3D](#).

4.22.2.3 void [SILL::PRINT](#) ([DATAFILE](#) & *out*) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Reimplemented from [SOURCE_BE3D](#).

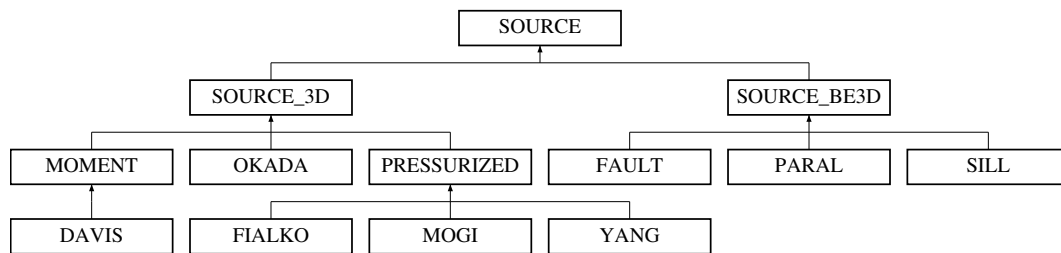
The documentation for this class was generated from the following files:

- SRC/[SILL.h](#)
- SRC/[SILL.cpp](#)

4.23 SOURCE Class Reference

```
#include <SOURCE.h>
```

Inheritance diagram for SOURCE:



Public Member Functions

- [SOURCE](#) ()
- [SOURCE](#) (string label_i, double x0_i, double y0_i, double z0_i)
- virtual [~SOURCE](#) ()
- virtual void [SET](#) (MEDIUM MEDIUM_PAR_i)
- virtual void [DISPLACEMENT](#) (double x, double y, double z, double U[])
- virtual void [DISPLACEMENT](#) (double r, double z, double &Ur, double &Uz)
- virtual void [STRAIN](#) (double x, double y, double z, double Strain[])
- virtual void [STRESS](#) (double x, double y, double z, double Stress[])
- virtual void [PRINT](#) ()
- virtual void [PRINT](#) (CONSOLE &out)
- virtual void [PRINT](#) (DATAFILE &out)=0
- virtual void [GET_label](#) (string &label_o)
- virtual void [GET_x0_y0_z0](#) (double &x0_o, double &y0_o, double &z0_o)
- virtual void [GET_MEDIUM_PAR](#) (MEDIUM &MEDIUM_PAR_o)
- virtual void [PUT_label](#) (string label_i)
- virtual void [PUT_x0_y0_z0](#) (double x0_i, double y0_i, double z0_i)
- virtual void [PUT_MEDIUM_PAR](#) (MEDIUM MEDIUM_PAR_i)

Protected Attributes

- string [label](#)
- double [x0](#)
- double [y0](#)
- double [z0](#)
- [MEDIUM MEDIUM_PAR](#)

4.23.1 Constructor & Destructor Documentation

4.23.1.1 [SOURCE::SOURCE](#) ()

4.23.1.2 [SOURCE::SOURCE](#) (string label_i, double x0_i, double y0_i, double z0_i) [inline]

4.23.1.3 [SOURCE::~~SOURCE](#) () [virtual]

4.23.2 Member Function Documentation

4.23.2.1 virtual void [SOURCE::DISPLACEMENT](#) (double x, double y, double z, double U[]) [inline],[virtual]

Computation of the displacement components ($U_x=U[0]$, $U_y=U[1]$, $U_z=U[2]$) at the observation point (x,y,z)

Reimplemented in [SOURCE_BE3D](#), [OKADA](#), [YANG](#), [FIALKO](#), [MOMENT](#), [DAVIS](#), and [MOGI](#).

4.23.2.2 `virtual void SOURCE::DISPLACEMENT (double r, double z, double & Ur, double & Uz) [inline],
[virtual]`

Computation of the displacement components (*Ur* radial component, *Uz* vertical component) at the observation point (*x,y,z*) (available only for axially symmetric sources)

Reimplemented in [FIALKO](#), and [MOGI](#).

4.23.2.3 `virtual void SOURCE::GET_label (string & label_o) [inline],[virtual]`

We retrieve the label associated to the source

4.23.2.4 `virtual void SOURCE::GET_MEDIUM_PAR (MEDIUM & MEDIUM_PAR_o) [inline],[virtual]`

We retrieve the MEDIUM_PAR object which describes the properties of the medium where the source is buried

4.23.2.5 `virtual void SOURCE::GET_x0_y0_z0 (double & x0_o, double & y0_o, double & z0_o) [inline],[virtual]`

We retrieve the coordinates of the source center (*x0,y0,z0*)

4.23.2.6 `virtual void SOURCE::PRINT () [inline],[virtual]`

Generation of a datafile and a gnuplot script suitable to obtain a graphic representation of the source model geometry

Reimplemented in [SOURCE_BE3D](#), [OKADA](#), [PARAL](#), [FAULT](#), and [SILL](#).

4.23.2.7 `virtual void SOURCE::PRINT (CONSOLE & out) [inline],[virtual]`

Print to standard output useful information about the source

Reimplemented in [SOURCE_BE3D](#), [OKADA](#), [PARAL](#), [YANG](#), [FIALKO](#), [FAULT](#), [MOMENT](#), [SILL](#), and [MOGI](#).

4.23.2.8 `virtual void SOURCE::PRINT (DATAFILE & out) [pure virtual]`

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface

Implemented in [SOURCE_BE3D](#), [OKADA](#), [PARAL](#), [YANG](#), [FIALKO](#), [FAULT](#), [MOMENT](#), [SILL](#), and [MOGI](#).

4.23.2.9 `virtual void SOURCE::PUT_label (string label_i) [inline],[virtual]`

We change the label associated to the source

4.23.2.10 `virtual void SOURCE::PUT_MEDIUM_PAR (MEDIUM MEDIUM_PAR_i) [inline],[virtual]`

We change the MEDIUM_PAR object which describes the properties of the medium where the source is buried

4.23.2.11 `virtual void SOURCE::PUT_x0_y0_z0 (double x0_i, double y0_i, double z0_i) [inline],[virtual]`

We change the coordinates of the source center (*x0,y0,z0*)

Reimplemented in [SOURCE_BE3D](#).

4.23.2.12 `virtual void SOURCE::SET (MEDIUM MEDIUM_PAR_i) [inline],[virtual]`

Association of the medium param with the source buried in the Half-space

4.23.2.13 `virtual void SOURCE::STRAIN (double x, double y, double z, double Strain[]) [inline],[virtual]`

Computation of the strain tensor components (S_{xx} =Strain[0], S_{yy} =Strain[1], S_{zz} =Strain[2], S_{xy} =Strain[3], S_{xz} =Strain[4], S_{yz} =Strain[5]) at the observation point (x,y,z)

Reimplemented in [SOURCE_BE3D](#), and [OKADA](#).

4.23.2.14 `virtual void SOURCE::STRESS (double x, double y, double z, double Stress[]) [inline],[virtual]`

Computation of the stress tensor components (S_{xx} =Stress[0], S_{yy} =Stress[1], S_{zz} =Stress[2], S_{xy} =Stress[3], S_{xz} =Stress[4], S_{yz} =Stress[5]) at the observation point (x,y,z)

Reimplemented in [SOURCE_BE3D](#), and [OKADA](#).

4.23.3 Member Data Documentation

4.23.3.1 `string SOURCE::label [protected]`

4.23.3.2 `MEDIUM SOURCE::MEDIUM_PAR [protected]`

4.23.3.3 `double SOURCE::x0 [protected]`

4.23.3.4 `double SOURCE::y0 [protected]`

4.23.3.5 `double SOURCE::z0 [protected]`

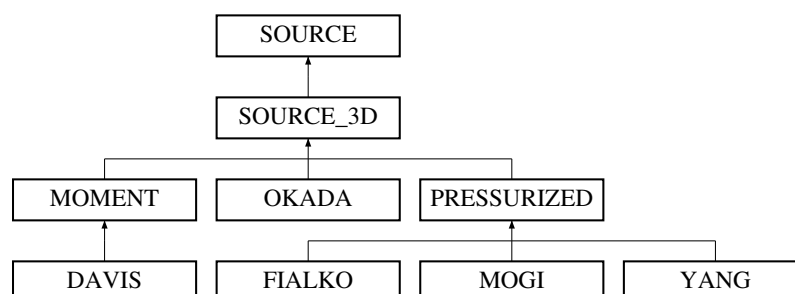
The documentation for this class was generated from the following files:

- SRC/[SOURCE.h](#)
- SRC/[SOURCE.cpp](#)

4.24 SOURCE_3D Class Reference

```
#include <SOURCE_3D.h>
```

Inheritance diagram for SOURCE_3D:



Public Member Functions

- [SOURCE_3D](#) ()
- [SOURCE_3D](#) (string *s_label*, double *x0*, double *y0*, double *z0*)
- virtual [~SOURCE_3D](#) ()

Additional Inherited Members

4.24.1 Constructor & Destructor Documentation

4.24.1.1 [SOURCE_3D::SOURCE_3D](#) ()

4.24.1.2 [SOURCE_3D::SOURCE_3D](#) (string *s_label*, double *x0*, double *y0*, double *z0*) [\[inline\]](#)

4.24.1.3 [SOURCE_3D::~~SOURCE_3D](#) () [\[virtual\]](#)

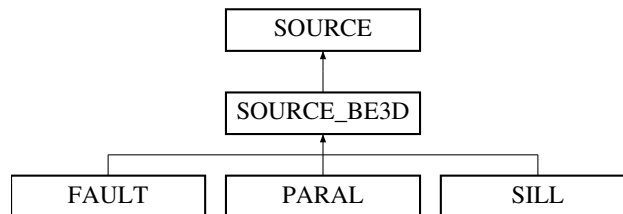
The documentation for this class was generated from the following files:

- SRC/[SOURCE_3D.h](#)
- SRC/[SOURCE_3D.cpp](#)

4.25 SOURCE_BE3D Class Reference

```
#include <SOURCE_BE3D.h>
```

Inheritance diagram for SOURCE_BE3D:



Public Member Functions

- [SOURCE_BE3D](#) ()
- [SOURCE_BE3D](#) (string *s_label*, double *x0*, double *y0*, double *z0*, double *phi_angle_i*, int *N*)
- virtual [~SOURCE_BE3D](#) ()
- virtual void [SET](#) ([MEDIUM](#) *MEDIUM_PAR_i*, int *option_flag*, double *INPUT_1*, double *INPUT_2*, double *INPUT_3*, int *FLAG_COMP_i*=1)
- void [DISPLACEMENT](#) (double *x*, double *y*, double *z*, double *U*[])
- void [STRAIN](#) (double *x*, double *y*, double *z*, double *Strain*[])
- void [STRESS](#) (double *x*, double *y*, double *z*, double *Stress*[])
- virtual void [SOLVE](#) ()
- virtual void [PRINT](#) ()
- virtual void [PRINT](#) ([CONSOLE](#) &out)
- virtual void [PRINT](#) ([DATAFILE](#) &out)
- void [GET_phi](#) (double *phi_o*)
- void [GET_NBE](#) (int *NBE_o*)
- void [GET_NDP](#) (int *NDP_o*)
- void [GET_FLAG_COMP](#) (int *FLAG_COMP_o*)
- void [PUT_x0_y0_z0](#) (double *x0_i*, double *y0_i*, double *z0_i*)

Protected Member Functions

- void [PRINT_GEOM_PROP](#) (ostream *out_stream)
- void [PRINT_STRESS_PROP](#) (ostream *stream)
- void [PRINT_BV_COMP](#) (ostream *stream)
- void [PRINT_MAIN_FEATURES](#) (ostream *out_stream, int flag_print=0)

Protected Attributes

- double [phi_degree](#)
- int [NBE](#)
- [BE3D](#) * [BE](#)
- int [NDP](#)
- int [FLAG_COMP](#)
- [GRID](#) [GRID_PAR](#)

4.25.1 Constructor & Destructor Documentation

4.25.1.1 [SOURCE_BE3D::SOURCE_BE3D](#) ()

4.25.1.2 [SOURCE_BE3D::SOURCE_BE3D](#) (string *s_label*, double *x0*, double *y0*, double *z0*, double *phi_angle_i*, int *N*)
[inline]

4.25.1.3 [SOURCE_BE3D::~~SOURCE_BE3D](#) () [virtual]

4.25.2 Member Function Documentation

4.25.2.1 void [SOURCE_BE3D::DISPLACEMENT](#) (double *x*, double *y*, double *z*, double *U*[]) [virtual]

Computation of the displacement components ($U_x=U[0]$, $U_y=U[1]$, $U_z=U[2]$) at the observation point (x,y,z)

Reimplemented from [SOURCE](#).

4.25.2.2 void [SOURCE_BE3D::GET_FLAG_COMP](#) (int *FLAG_COMP_o*) [inline]

4.25.2.3 void [SOURCE_BE3D::GET_NBE](#) (int *NBE_o*) [inline]

With this method, we query the object in order to retrieve the NBE parameter associated to the source.

4.25.2.4 void [SOURCE_BE3D::GET_NDP](#) (int *NDP_o*) [inline]

With this method, we query the object in order to retrieve the NBE parameter associated to the source.

4.25.2.5 void [SOURCE_BE3D::GET_phi](#) (double *phi_o*) [inline]

With this method, we query the object in order to retrieve the phi angle associated to the source.

4.25.2.6 void [SOURCE_BE3D::PRINT](#) () [virtual]

Generation of a datafile and a gnuplot script suitable to obtain a graphic representation of the source model geometry

Reimplemented from [SOURCE](#).

Reimplemented in [PARAL](#), [FAULT](#), and [SILL](#).

4.25.2.7 void SOURCE_BE3D::PRINT (**CONSOLE** & *out*) [virtual]

Print to standard output useful information about the source

Reimplemented from [SOURCE](#).

Reimplemented in [PARAL](#), [FAULT](#), and [SILL](#).

4.25.2.8 void SOURCE_BE3D::PRINT (**DATAFILE** & *out*) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface.

Implements [SOURCE](#).

Reimplemented in [PARAL](#), [FAULT](#), and [SILL](#).

4.25.2.9 void SOURCE_BE3D::PRINT_BV_COMP (ostream * *stream*) [protected]

4.25.2.10 void SOURCE_BE3D::PRINT_GEOM_PROP (ostream * *out_stream*) [protected]

4.25.2.11 void SOURCE_BE3D::PRINT_MAIN_FEATURES (ostream * *out_stream*, int *flag_print* = 0) [protected]

4.25.2.12 void SOURCE_BE3D::PRINT_STRESS_PROP (ostream * *stream*) [protected]

4.25.2.13 void SOURCE_BE3D::PUT_x0_y0_z0 (double *x0_i*, double *y0_i*, double *z0_i*) [virtual]

We change the coordinates of the source center (x0,y0,z0)

Reimplemented from [SOURCE](#).

4.25.2.14 virtual void SOURCE_BE3D::SET (**MEDIUM** *MEDIUM_PAR_i*, int *option_flag*, double *INPUT_1*, double *INPUT_2*, double *INPUT_3*, int *FLAG_COMP_i* = 1) [inline],[virtual]

Association of the medium param with the source buried in the Half-space This method uses the integer variable *option_flag* to choose the type of input expected:

- *option_flag* = 1, we consider the direct problem in fracture theory; a uniform slip is assigned over the source boundaries, with strike-slip component equal to *INPUT_1*, with dip-slip component equal to *INPUT_2* and with tensile component with strike-slip component equal to *INPUT_3*;
- *option_flag* = 2, we consider the inverse problem in fracture theory, uniform boundary conditions are assigned over source boundaries. The optional variable *FLAG_COMP_i* can be specified in order to select the Okada implementation adopted:
- *FLAG_COMP_i* = 1, the implementation given by Ferrari et Al. (default option);
- *FLAG_COMP_i* = 2, the original implementation by Okada (dc3d.f code)

Reimplemented in [PARAL](#).

4.25.2.15 void SOURCE_BE3D::SOLVE () [virtual]

Reimplemented in [PARAL](#).

4.25.2.16 void SOURCE_BE3D::STRAIN (double x, double y, double z, double Strain[]) [virtual]

Computation of the strain tensor components (Sxx=Strain[0], Syy=Strain[1], Szz=Strain[2], Sxy=Strain[3], Sxz=Strain[4], Syz=Strain[5]) at the observation point (x,y,z)

Reimplemented from [SOURCE](#).

4.25.2.17 void SOURCE_BE3D::STRESS (double x, double y, double z, double Stress[]) [virtual]

Computation of the stress tensor components (Sxx=Stress[0], Syy=Stress[1], Szz=Stress[2], Sxy=Stress[3], Sxz=Stress[4], Syz=Stress[5]) at the observation point (x,y,z) SOLVE is the method used to calculate the solution relative to the inverse problem in fracture theory. SOLVE implements the discontinuity element method (BEM)

Reimplemented from [SOURCE](#).

4.25.3 Member Data Documentation

4.25.3.1 BE3D* SOURCE_BE3D::BE [protected]

4.25.3.2 int SOURCE_BE3D::FLAG_COMP [protected]

4.25.3.3 GRID SOURCE_BE3D::GRID_PAR [protected]

4.25.3.4 int SOURCE_BE3D::NBE [protected]

4.25.3.5 int SOURCE_BE3D::NDP [protected]

4.25.3.6 double SOURCE_BE3D::phi_degree [protected]

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

- SRC/[SOURCE_BE3D.h](#)
- SRC/[SOURCE_BE3D.cpp](#)

4.26 source_param Struct Reference

```
#include <foo.h>
```

Public Attributes

- double [Delta_P](#)
- double [c1](#)
- double [h_1](#)
- double [h_2](#)
- double [h_3](#)
- double [max_side](#)
- int [type](#)

4.26.1 Member Data Documentation

4.26.1.1 double source_param::c1

4.26.1.2 double source_param::Delta_P

4.26.1.3 double source_param::h_1

4.26.1.4 double source_param::h_2

4.26.1.5 double source_param::h_3

4.26.1.6 double source_param::max_side

4.26.1.7 int source_param::type

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

- SRC/[foo.h](#)

4.27 struct_var Struct Reference

```
#include <foo.h>
```

Public Attributes

- double [j1A](#)
- double [J1A](#)
- double [j1B](#)
- double [j1C](#)
- double [j2A](#)
- double [J2A](#)
- double [j2B](#)
- double [j2C](#)
- double [j3A](#)
- double [J3A](#)
- double [j3B](#)
- double [j3C](#)
- double [k1A](#)
- double [K1A](#)
- double [k1B](#)
- double [k1C](#)
- double [k2A](#)
- double [K2A](#)
- double [k2B](#)
- double [k2C](#)
- double [k3A](#)
- double [K3A](#)
- double [k3B](#)
- double [k3C](#)
- double [l1A](#)
- double [L1A](#)
- double [l1B](#)
- double [u1C](#)
- double [l1C](#)
- double [l2A](#)
- double [L2A](#)

- double [l2B](#)
- double [u2C](#)
- double [l2C](#)
- double [l3A](#)
- double [L3A](#)
- double [l3B](#)
- double [u3C](#)
- double [l3C](#)
- double [Ux_x](#)
- double [Uy_y](#)
- double [Uz_z](#)
- double [Ux_y](#)
- double [Uy_x](#)
- double [Ux_z](#)
- double [Uz_x](#)
- double [Uy_z](#)
- double [Uz_y](#)
- double [u1A](#)
- double [U1A](#)
- double [u1B](#)
- double [u2A](#)
- double [U2A](#)
- double [u2B](#)
- double [u3A](#)
- double [U3A](#)
- double [u3B](#)
- double [Ux](#)
- double [Uy](#)
- double [Uz](#)

4.27.1 Member Data Documentation

4.27.1.1 double struct_var::j1A

4.27.1.2 double struct_var::J1A

4.27.1.3 double struct_var::j1B

4.27.1.4 double struct_var::j1C

4.27.1.5 double struct_var::j2A

4.27.1.6 double struct_var::J2A

4.27.1.7 double struct_var::j2B

4.27.1.8 double struct_var::j2C

4.27.1.9 double struct_var::j3A

4.27.1.10 double struct_var::J3A

4.27.1.11 double struct_var::j3B

4.27.1.12 double struct_var::j3C

4.27.1.13 double struct_var::k1A

4.27.1.14 double struct_var::K1A

4.27.1.15 double struct_var::k1B

4.27.1.16 double struct_var::k1C

4.27.1.17 double struct_var::k2A

4.27.1.18 double struct_var::K2A

4.27.1.19 double struct_var::k2B

4.27.1.20 double struct_var::k2C

4.27.1.21 double struct_var::k3A

4.27.1.22 double struct_var::K3A

4.27.1.23 double struct_var::k3B

4.27.1.24 double struct_var::k3C

4.27.1.25 double struct_var::l1A

4.27.1.26 double struct_var::L1A

4.27.1.27 double struct_var::l1B

4.27.1.28 double struct_var::l1C

4.27.1.29 double struct_var::l2A

4.27.1.30 double struct_var::L2A

4.27.1.31 double struct_var::l2B

4.27.1.32 double struct_var::l2C

4.27.1.33 double struct_var::l3A

4.27.1.34 double struct_var::L3A

4.27.1.35 double struct_var::l3B

4.27.1.36 double struct_var::l3C

4.27.1.37 double struct_var::u1A

4.27.1.38 double struct_var::U1A

4.27.1.39 double struct_var::u1B

4.27.1.40 double struct_var::u1C
4.27.1.41 double struct_var::U2A
4.27.1.42 double struct_var::u2A
4.27.1.43 double struct_var::u2B
4.27.1.44 double struct_var::u2C
4.27.1.45 double struct_var::U3A
4.27.1.46 double struct_var::u3A
4.27.1.47 double struct_var::u3B
4.27.1.48 double struct_var::u3C
4.27.1.49 double struct_var::Ux
4.27.1.50 double struct_var::Ux_x
4.27.1.51 double struct_var::Ux_y
4.27.1.52 double struct_var::Ux_z
4.27.1.53 double struct_var::Uy
4.27.1.54 double struct_var::Uy_x
4.27.1.55 double struct_var::Uy_y
4.27.1.56 double struct_var::Uy_z
4.27.1.57 double struct_var::Uz
4.27.1.58 double struct_var::Uz_x
4.27.1.59 double struct_var::Uz_y
4.27.1.60 double struct_var::Uz_z

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

- SRC/[foo.h](#)

4.28 subsource_param_tj Struct Reference

```
#include <foo.h>
```

Public Attributes

- double [x_cc](#)
- double [y_cc](#)

- double [c_cc](#)
- double [x_pc](#)
- double [y_pc](#)
- double [c_pc](#)
- double [delta_gradi](#)
- double [L](#)
- double [W](#)
- int [flag](#)
- int [tj](#)
- int [l2](#)

4.28.1 Member Data Documentation

4.28.1.1 double subsource_param_tj::c_cc

4.28.1.2 double subsource_param_tj::c_pc

4.28.1.3 double subsource_param_tj::delta_gradi

4.28.1.4 int subsource_param_tj::flag

4.28.1.5 int subsource_param_tj::l2

4.28.1.6 double subsource_param_tj::L

4.28.1.7 int subsource_param_tj::tj

4.28.1.8 double subsource_param_tj::W

4.28.1.9 double subsource_param_tj::x_cc

4.28.1.10 double subsource_param_tj::x_pc

4.28.1.11 double subsource_param_tj::y_cc

4.28.1.12 double subsource_param_tj::y_pc

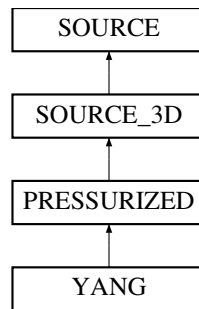
The documentation for this struct was generated from the following file:

- [SRC/foo.h](#)

4.29 YANG Class Reference

```
#include <YANG.h>
```

Inheritance diagram for YANG:



Public Member Functions

- [YANG](#) ()
- [YANG](#) (string label_i, double x0_i, double y0_i, double z0_i, double a_i, double b_i, double phi_i, double theta_i)
- virtual [~YANG](#) ()
- void [SET](#) ([MEDIUM](#) MEDIUM_PAR_i, double P_i)
- void [DISPLACEMENT](#) (double x, double y, double z, double U[])
- void [PRINT](#) ([CONSOLE](#) &out)
- void [PRINT](#) ([DATAFILE](#) &out)

Additional Inherited Members

4.29.1 Constructor & Destructor Documentation

4.29.1.1 YANG::YANG ()

4.29.1.2 `YANG::YANG (string label_i, double x0_i, double y0_i, double z0_i, double a_i, double b_i, double phi_i, double theta_i) [inline]`

4.29.1.3 YANG::~~YANG () [virtual]

4.29.2 Member Function Documentation

4.29.2.1 void YANG::DISPLACEMENT (double x, double y, double z, double U[]) [virtual]

Computation of the displacement components ($U_x=U[0]$, $U_y=U[1]$, $U_z=U[2]$) at the observation point (x,y,z)

Reimplemented from [SOURCE](#).

4.29.2.2 void YANG::PRINT ([CONSOLE](#) & out) [virtual]

Print to standard output useful information about the source

Reimplemented from [SOURCE](#).

4.29.2.3 void YANG::PRINT ([DATAFILE](#) & out) [virtual]

Generation of a datafile and a gnuplot script. According to the [DATAFILE](#) object passed to PRINT, we obtain displacement and/or strain and/or stress maps at the free surface.

Implements [SOURCE](#).

4.29.2.4 void YANG::SET (MEDIUM MEDIUM_PAR_i, double P_i) [virtual]

Association of the medium param with the source buried in the Half-space. It also assigns the overpressure.

Implements [PRESSURIZED](#).

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

- SRC/[YANG.h](#)
- SRC/[YANG.cpp](#)

Chapter 5

File Documentation

5.1 SRC/BE3D.cpp File Reference

```
#include <cmath>
#include <iostream>
#include "BE3D.h"
#include "MEDIUM.h"
#include "Utilities.h"
```

Functions

- void **DC3D** (float **alpha**, float **X**, float **Y**, float **Z**, float **DEPTH**, float **DIP**, float **AL1**, float **AL2**, float **AW1**, float **AW2**, float **DISL1**, float **DISL2**, float **DISL3**, float **&UX**, float **&UY**, float **&UZ**, float **&UXX**, float **&UYX**, float **&UZX**, float **&UXY**, float **&UYX**, float **&UZY**, float **&UXZ**, float **&UYZ**, float **&UZZ**, int **&IRET**)

Variables

- const double **EPS** =1e-6
- double **SIN_DELTA**
- double **COS_DELTA**
- double **SIN_DELTA_2**
- double **COS_DELTA_2**
- double **alpha**
- double **coef_alpha_1**
- double **coef_alpha_2**
- double **p**
- double **q**
- double **Y**
- double **D**
- double **RpD**
- double **RpD2**
- double **Xi2**
- double **Eta2**
- double **q2**
- double **q3**
- double **Y2**
- double **D2**
- double **R**

- double [R2](#)
- double [R3](#)
- double [R5](#)
- double [D_11](#)
- double [X_11](#)
- double [X_32](#)
- double [X_53](#)
- double [Y_11](#)
- double [Y_32](#)
- double [Y_53](#)
- double [Y_0](#)
- double [E](#)
- double [F](#)
- double [G](#)
- double [H](#)
- double [Ep](#)
- double [Fp](#)
- double [Gp](#)
- double [Hp](#)
- double [Teta](#)
- double [X](#)
- double [Sing_iii](#)
- double [Sing_iv](#)
- double [I_1](#)
- double [I_2](#)
- double [I_3](#)
- double [I_4](#)
- double [J_1](#)
- double [J_2](#)
- double [J_3](#)
- double [J_4](#)
- double [J_5](#)
- double [J_6](#)
- double [K_1](#)
- double [K_2](#)
- double [K_3](#)
- double [K_4](#)
- double [C](#)
- double [h](#)
- double [Z_32](#)
- double [Z_53](#)
- double [Z_0](#)
- double [CR](#)
- double [P](#)
- double [Q](#)
- double [Pp](#)
- double [Qp](#)
- double [qR](#)
- double [CDR](#)
- double [YY0](#)
- const float [pi](#) =acos(-1)
- const float [coefpi](#) =1.E0/(2.E0*pi)

5.1.1 Function Documentation

5.1.1.1 void DC3D (float *alpha*, float *X*, float *Y*, float *Z*, float *DEPTH*, float *DIP*, float *AL1*, float *AL2*, float *AW1*, float *AW2*, float *DISL1*, float *DISL2*, float *DISL3*, float & *UX*, float & *UY*, float & *UZ*, float & *UXX*, float & *UYX*, float & *UZX*, float & *UXY*, float & *UYZ*, float & *UZY*, float & *UXZ*, float & *UYZ*, float & *UZZ*, int & *IRET*)

5.1.2 Variable Documentation

5.1.2.1 double *alpha*

5.1.2.2 double *C*

5.1.2.3 double *CDR*

5.1.2.4 double *coef_alpha_1*

5.1.2.5 double *coef_alpha_2*

5.1.2.6 const float *coefpi* =1.E0/(2.E0*pi)

5.1.2.7 double *COS_DELTA*

5.1.2.8 double *COS_DELTA_2*

5.1.2.9 double *CR*

5.1.2.10 double *D*

5.1.2.11 double *D2*

5.1.2.12 double *D_11*

5.1.2.13 double *E*

5.1.2.14 double *Ep*

5.1.2.15 const double *EPS* =1e-6

5.1.2.16 double *Eta2*

5.1.2.17 void *F*

5.1.2.18 double *Fp*

5.1.2.19 double *G*

5.1.2.20 double *Gp*

5.1.2.21 double *H*

5.1.2.22 double *h*

5.1.2.23 double *Hp*

5.1.2.24 double *I_1*

5.1.2.25 double I_2

5.1.2.26 double I_3

5.1.2.27 double I_4

5.1.2.28 double J_1

5.1.2.29 double J_2

5.1.2.30 double J_3

5.1.2.31 double J_4

5.1.2.32 double J_5

5.1.2.33 double J_6

5.1.2.34 double K_1

5.1.2.35 double K_2

5.1.2.36 double K_3

5.1.2.37 double K_4

5.1.2.38 double p

5.1.2.39 double P

5.1.2.40 const float pi =acos(-1)

5.1.2.41 double Pp

5.1.2.42 double q

5.1.2.43 vector< double > Q

5.1.2.44 double q2

5.1.2.45 double q3

5.1.2.46 double Qp

5.1.2.47 double qR

5.1.2.48 double R

5.1.2.49 double R2

5.1.2.50 double R3

5.1.2.51 double R5

5.1.2.52 double RpD

5.1.2.53 double RpD2

5.1.2.54 double SIN_DELTA

5.1.2.55 double SIN_DELTA_2

5.1.2.56 double Sing_iii

5.1.2.57 double Sing_iv

5.1.2.58 double Teta

5.1.2.59 double X

5.1.2.60 double X_11

5.1.2.61 double X_32

5.1.2.62 double X_53

5.1.2.63 double Xi2

5.1.2.64 double Y

5.1.2.65 double Y2

5.1.2.66 double Y_0

5.1.2.67 double Y_11

5.1.2.68 double Y_32

5.1.2.69 double Y_53

5.1.2.70 double YY0

5.1.2.71 double Z_0

5.1.2.72 double Z_32

5.1.2.73 double Z_53

5.2 SRC/BE3D.h File Reference

```
#include "BE3D_PARAM.h"
#include "foo.h"
#include "MEDIUM.h"
```

Classes

- class [BE3D](#)

5.3 SRC/BE3D_GEOM_PARAM.h File Reference

```
#include <cmath>
```

Classes

- class [BE3D_GEOM_PARAM](#)

5.4 SRC/BE3D_PARAM.h File Reference

```
#include "BE3D_GEOM_PARAM.h"  
#include "BE3D_STRESS_PARAM.h"
```

Classes

- class [BE3D_PARAM](#)

5.5 SRC/BE3D_STRESS_PARAM.h File Reference

Classes

- class [BE3D_STRESS_PARAM](#)

5.6 SRC/CONSOLE.cpp File Reference

```
#include <iostream>  
#include <fstream>  
#include <string>  
#include "CONSOLE.h"
```

5.7 SRC/CONSOLE.h File Reference

```
#include <fstream>  
#include <string>  
#include "OUTPUT.h"
```

Classes

- class [CONSOLE](#)

5.8 SRC/DATAFILE.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include "DATAFILE.h"
```

5.9 SRC/DATAFILE.h File Reference

```
#include <fstream>
#include <string>
#include <vector>
#include "OUTPUT.h"
```

Classes

- class [DATAFILE](#)

5.10 SRC/DAVIS.cpp File Reference

```
#include "DAVIS.h"
#include <iomanip>
#include "Utilities.h"
```

Functions

- void [A2M](#) (float *nu*, float *a*, float *b*, float *c*, float &*M1*, float &*M2*, float &*M3*)

Variables

- double [pi](#) =acos(-1)

5.10.1 Function Documentation

5.10.1.1 void [A2M](#) (float *nu*, float *a*, float *b*, float *c*, float & *M1*, float & *M2*, float & *M3*)

5.10.2 Variable Documentation

5.10.2.1 double [pi](#) =acos(-1)

5.11 SRC/DAVIS.h File Reference

```
#include "MOMENT.h"
```

Classes

- class [DAVIS](#)

Functions

- float [rd](#) (float *x*, float *y*, float *z*)
- void [A2M](#) (float *nu*, float *a*, float *b*, float *c*, float &*M1*, float &*M2*, float &*M3*)
- void [sgesv_](#) (const int **N*, const int **nrhs*, float **A*, const int **lda*, int **ipiv*, float **b*, const int **ldb*, int **info*)

Variables

- const double [TABLE](#) [78][5]
- const double [TABLE_range](#) [251][2]

5.11.1 Function Documentation

5.11.1.1 void [A2M](#) (float *nu*, float *a*, float *b*, float *c*, float & *M1*, float & *M2*, float & *M3*)

5.11.1.2 float [rd](#) (float *x*, float *y*, float *z*)

5.11.1.3 void [sgesv_](#) (const int * *N*, const int * *nrhs*, float * *A*, const int * *lda*, int * *ipiv*, float * *b*, const int * *ldb*, int * *info*)

5.11.2 Variable Documentation

5.11.2.1 const double [TABLE](#)[78][5]

5.11.2.2 const double [TABLE_range](#)[251][2]

5.12 SRC/FAULT.cpp File Reference

```
#include "FAULT.h"
#include <cmath>
#include <string>
#include <iomanip>
#include "BE3D.h"
#include "PRINT_TEMPLATE.h"
```

5.13 SRC/FAULT.h File Reference

```
#include <typeinfo>
#include "BE3D.h"
#include "CONSOLE.h"
#include "DATAFILE.h"
#include "GRID.h"
#include "SOURCE_BE3D.h"
```

Classes

- class [FAULT](#)

5.14 SRC/FIALKO.cpp File Reference

```
#include <iostream>
#include <cmath>
#include <vector>
#include <algorithm>
#include "FIALKO.h"
#include "MEDIUM.h"
#include "PRINT_TEMPLATE.h"
```

Functions

- `vector< double > fpkernel (double h, double ti, vector< double > r, int n)`
- `vector< double > KG (vector< double > s, double p)`
- `vector< double > KERN (vector< double > w, double p)`
- `vector< double > Q (double h, vector< double > t, double r, int n)`

5.14.1 Function Documentation

5.14.1.1 `vector< double > fpkernel (double h, double ti, vector< double > r, int n)`

5.14.1.2 `vector< double > KERN (vector< double > w, double p)`

5.14.1.3 `vector< double > KG (vector< double > s, double p)`

5.14.1.4 `vector<double> Q (double h, vector< double > t, double r, int n)`

5.15 SRC/FIALKO.h File Reference

```
#include <vector>
#include "PRESSURIZED.h"
```

Classes

- class `FIALKO`

5.16 SRC/foo.h File Reference

Classes

- struct `source_param`
- struct `medium_param`
- struct `discretization_param`
- struct `obs_point_ti`
- struct `subsource_param_tj`
- struct `struct_var`
- struct `gravity_grid_param`

Variables

- const int [max_dim](#) =27000

5.16.1 Variable Documentation

5.16.1.1 const int max_dim =27000

5.17 SRC/GRID.h File Reference

Classes

- class [GRID](#)

5.18 SRC/MEDIUM.cpp File Reference

```
#include "MEDIUM.h"
```

5.19 SRC/MEDIUM.h File Reference

Classes

- class [MEDIUM](#)

5.20 SRC/MOGL.cpp File Reference

```
#include "MOGL.h"  
#include <cmath>  
#include "PRINT_TEMPLATE.h"
```

5.21 SRC/MOGL.h File Reference

```
#include "PRESSURIZED.h"
```

Classes

- class [MOGL](#)

5.22 SRC/MOMENT.cpp File Reference

```
#include "MOMENT.h"  
#include "cmath"  
#include "PRINT_TEMPLATE.h"
```

Functions

- double [G_hs](#) (int *tj*, int *tk*, int *ti*, double *X*, double *Y*, double *xi*, double *mu*, double *nu*)

5.22.1 Function Documentation

5.22.1.1 double [G_hs](#) (int *tj*, int *tk*, int *ti*, double *X*, double *Y*, double *xi*, double *mu*, double *nu*)

5.23 SRC/MOMENT.h File Reference

```
#include "SOURCE_3D.h"
```

Classes

- class [MOMENT](#)

5.24 SRC/OKADA.cpp File Reference

```
#include "OKADA.h"  
#include <math.h>  
#include "PRINT_TEMPLATE.h"
```

5.25 SRC/OKADA.h File Reference

```
#include "string.h"  
#include "BE3D.h"  
#include "SOURCE_3D.h"
```

Classes

- class [OKADA](#)
Brief description. Brief description continued.

5.26 SRC/OUTPUT.h File Reference

```
#include <fstream>
```

Classes

- class [OUTPUT](#)
Brief description. Brief description continued.

5.27 SRC/PARAL.cpp File Reference

```
#include <cmath>
#include <iomanip>
#include <typeinfo>
#include "PARAL.h"
#include "GRID.h"
#include "PRINT_TEMPLATE.h"
#include "foo.h"
#include "MEDIUM.h"
```

Functions

- void [dgesv_](#) (const int *N, const int *nrhs, double *A, const int *lda, int *ipiv, double *b, const int *ldb, int *info)
- void [F](#) (int &N_a, int &N_b, int NUM, double h_a, double h_b, double passo)
- void [PARAL_f_GEOM_PROP](#) (ostream *out_stream)
- void [f_STRESS_PROP](#) (ostream *stream)
- void [f_DISL_PROP](#) (ostream *stream)
- void [f_BV_PROP](#) (ostream *stream)

5.27.1 Function Documentation

5.27.1.1 void [dgesv_](#) (const int * *N*, const int * *nrhs*, double * *A*, const int * *lda*, int * *ipiv*, double * *b*, const int * *ldb*, int * *info*)

5.27.1.2 void [F](#) (int & *N_a*, int & *N_b*, int *NUM*, double *h_a*, double *h_b*, double *passo*)

5.27.1.3 void [f_BV_PROP](#) (ostream * *stream*)

5.27.1.4 void [f_DISL_PROP](#) (ostream * *stream*)

5.27.1.5 void [f_STRESS_PROP](#) (ostream * *stream*)

5.27.1.6 void [PARAL_f_GEOM_PROP](#) (ostream * *out_stream*)

5.28 SRC/PARAL.h File Reference

```
#include "cmath"
#include "BE3D.h"
#include "CONSOLE.h"
#include "DATAFILE.h"
#include "SOURCE_BE3D.h"
```

Classes

- class [PARAL](#)

Functions

- void [dgesv_](#) (const int *N, const int *nrhs, double *A, const int *lda, int *ipiv, double *b, const int *ldb, int *info)

5.28.1 Function Documentation

5.28.1.1 `void dgesv_ (const int * N, const int * nrhs, double * A, const int * lda, int * ipiv, double * b, const int * ldb, int * info)`

5.29 SRC/PRESSURIZED.cpp File Reference

```
#include "PRESSURIZED.h"
```

5.30 SRC/PRESSURIZED.h File Reference

```
#include "SOURCE_3D.h"
```

Classes

- class [PRESSURIZED](#)

5.31 SRC/PRINT_TEMPLATE.h File Reference

```
#include <string>
#include <iomanip>
#include <stdlib.h>
#include <iostream>
#include <typeinfo>
#include <ctime>
#include "Utilities.h"
#include "CONSOLE.h"
```

Functions

- template<class TSOURCE >
void [print_map_displ](#) (TSOURCE source, string filename, int N_1, int N_2, double DX1, double DX2)
- template<class TSOURCE >
void [print_displ_script](#) (TSOURCE source, string filename, double DX1, double DX2, int type)
- template<class TSOURCE >
void [print_map_strain](#) (TSOURCE source, string filename, int N_1, int N_2, double DX1, double DX2)
- template<class TSOURCE >
void [print_strain_script](#) (TSOURCE source, string filename, double DX1, double DX2, int type, int int_shift=0)
- template<class TSOURCE >
void [print_map_stress](#) (TSOURCE source, string filename, int N_1, int N_2, double DX1, double DX2)
- template<class TSOURCE >
void [print_stress_script](#) (TSOURCE source, string filename, double DX1, double DX2, int type, int int_shift=0)
- template<class TSOURCE >
void [print_maps](#) (TSOURCE source, string filename, int N_1, int N_2, double DX1, double DX2)
- template<class TSOURCE >
void [print_map_radial_displ](#) (TSOURCE source, string filename, int N_r, double r_i, double r_f)
- template<class TSOURCE >
void [print_radial_displ_script](#) (TSOURCE source, string filename, double r_i, double r_f)

5.31.1 Function Documentation

- 5.31.1.1 `template<class TSOURCE > void print_displ_script (TSOURCE source, string filename, double DX1, double DX2, int type)`
- 5.31.1.2 `template<class TSOURCE > void print_map_displ (TSOURCE source, string filename, int N_1, int N_2, double DX1, double DX2)`
- 5.31.1.3 `template<class TSOURCE > void print_map_radial_displ (TSOURCE source, string filename, int N_r, double r_i, double r_f)`
- 5.31.1.4 `template<class TSOURCE > void print_map_strain (TSOURCE source, string filename, int N_1, int N_2, double DX1, double DX2)`
- 5.31.1.5 `template<class TSOURCE > void print_map_stress (TSOURCE source, string filename, int N_1, int N_2, double DX1, double DX2)`
- 5.31.1.6 `template<class TSOURCE > void print_maps (TSOURCE source, string filename, int N_1, int N_2, double DX1, double DX2)`
- 5.31.1.7 `template<class TSOURCE > void print_radial_displ_script (TSOURCE source, string filename, double r_i, double r_f)`
- 5.31.1.8 `template<class TSOURCE > void print_strain_script (TSOURCE source, string filename, double DX1, double DX2, int type, int int_shift = 0)`
- 5.31.1.9 `template<class TSOURCE > void print_stress_script (TSOURCE source, string filename, double DX1, double DX2, int type, int int_shift = 0)`

5.32 SRC/SILL.cpp File Reference

```
#include "SILL.h"
#include <iomanip>
#include "PRINT_TEMPLATE.h"
```

5.33 SRC/SILL.h File Reference

```
#include "SOURCE_BE3D.h"
```

Classes

- class [SILL](#)

5.34 SRC/SOURCE.cpp File Reference

```
#include "SOURCE.h"
```

5.35 SRC/SOURCE.h File Reference

```
#include <string>
#include <iostream>
#include <stdlib.h>
#include <iomanip>
#include "CONSOLE.h"
#include "DATAFILE.h"
#include "MEDIUM.h"
```

Classes

- class [SOURCE](#)

5.36 SRC/SOURCE_3D.cpp File Reference

```
#include "SOURCE_3D.h"
```

5.37 SRC/SOURCE_3D.h File Reference

```
#include "SOURCE.h"
```

Classes

- class [SOURCE_3D](#)

5.38 SRC/SOURCE_BE3D.cpp File Reference

```
#include "SOURCE_BE3D.h"
#include <iomanip>
#include <stdlib.h>
#include <iostream>
#include <sstream>
#include <cmath>
#include "PRINT_TEMPLATE.h"
```

Functions

- void [dgesv_](#) (const int *N, const int *nrhs, double *A, const int *lda, int *ipiv, double *b, const int *ldb, int *info)

5.38.1 Function Documentation

5.38.1.1 void [dgesv_](#) (const int * *N*, const int * *nrhs*, double * *A*, const int * *lda*, int * *ipiv*, double * *b*, const int * *ldb*, int * *info*)

5.39 SRC/SOURCE_BE3D.h File Reference

```
#include "BE3D.h"
#include "GRID.h"
#include "SOURCE.h"
```

Classes

- class [SOURCE_BE3D](#)

5.40 SRC/Utilities.cpp File Reference

```
#include <stdio.h>
#include <iostream>
#include <fstream>
#include <gsl/gsl_math.h>
#include <gsl/gsl_eigen.h>
#include "cmath"
#include "Utilities.h"
```

Functions

- void [Cycle_counter](#) (int jf, int nsg, int c[])
- void [gsl_eigenvalueproblem](#) (int dimension, double eigenvalues[], double Matrix[])
- void [ref_func](#) (double &x, double &y, double &z, double X1, double X2, double X3_plane, int flag_plane)
- float [rd](#) (float x, float y, float z)
- void [dc3d_](#) (float *alpha, float *X, float *Y, float *Z, float *DEPTH, float *DIP, float *AL1, float *AL2, float *AW1, float *AW2, float *DISL1, float *DISL2, float *DISL3, float *UX, float *UY, float *UZ, float *UXX, float *UYX, float *UZX, float *UXY, float *UYU, float *UZY, float *UXZ, float *UYZ, float *UZZ, int *IRET)
- void [DC3D](#) (float alpha, float X, float Y, float Z, float DEPTH, float DIP, float AL1, float AL2, float AW1, float AW2, float DISL1, float DISL2, float DISL3, float &UX, float &UY, float &UZ, float &UXX, float &UYX, float &UZX, float &UXY, float &UYU, float &UZY, float &UXZ, float &UYZ, float &UZZ, int &IRET)

5.40.1 Function Documentation

5.40.1.1 void [Cycle_counter](#) (int jf, int nsg, int c[])

5.40.1.2 void [DC3D](#) (float alpha, float X, float Y, float Z, float DEPTH, float DIP, float AL1, float AL2, float AW1, float AW2, float DISL1, float DISL2, float DISL3, float &UX, float &UY, float &UZ, float &UXX, float &UYX, float &UZX, float &UXY, float &UYU, float &UZY, float &UXZ, float &UYZ, float &UZZ, int &IRET)

5.40.1.3 void [dc3d_](#) (float *alpha, float *X, float *Y, float *Z, float *DEPTH, float *DIP, float *AL1, float *AL2, float *AW1, float *AW2, float *DISL1, float *DISL2, float *DISL3, float *UX, float *UY, float *UZ, float *UXX, float *UYX, float *UZX, float *UXY, float *UYU, float *UZY, float *UXZ, float *UYZ, float *UZZ, int *IRET)

5.40.1.4 void [gsl_eigenvalueproblem](#) (int dimension, double eigenvalues[], double Matrix[])

5.40.1.5 float [rd](#) (float x, float y, float z)

5.40.1.6 void [ref_func](#) (double &x, double &y, double &z, double X1, double X2, double X3_plane, int flag_plane)

5.41 SRC/Utilities.h File Reference

```
#include "cmath"
```

Macros

- `#define ERRRTOL 0.05`
- `#define TINY 1.0e-25`
- `#define BIG 4.5e21`
- `#define C1 (3.0/14.0)`
- `#define C2 (1.0/6.0)`
- `#define C3 (9.0/22.0)`
- `#define C4 (3.0/26.0)`
- `#define C5 (0.25*C3)`
- `#define C6 (1.5*C4)`
- `#define FMAX(a, b)`

Functions

- `template<class type >`
`void Rotazione_coord (type &x, type &y, int flag)`
- `template<class type >`
`void Rotazione_coord (type &x, type &y, double phi_degree)`
- `template<class type >`
`void Rotazione_vettore (type V[], double phi_degree)`
- `template<class type >`
`void Rotazione_tensore (type S[], double phi_degree)`
- `float rd (float x, float y, float z)`
- `void Cycle_counter (int jf, int nsg, int c[])`
- `void gsl_eigenvalueproblem (int dimension, double eigenvalues[], double Matrix[])`
- `void ref_func (double &x, double &y, double &z, double X1, double X2, double X3_plane, int flag_plane)`
- `void DC3D (float alpha, float X, float Y, float Z, float DEPTH, float DIP, float AL1, float AL2, float AW1, float AW2, float DISL1, float DISL2, float DISL3, float &UX, float &UY, float &UZ, float &UXX, float &UYX, float &UXZ, float &UXY, float &UYZ, float &UZZ, int &IRET, int CASE)`

5.41.1 Macro Definition Documentation

5.41.1.1 `#define BIG 4.5e21`

5.41.1.2 `#define C1 (3.0/14.0)`

5.41.1.3 `#define C2 (1.0/6.0)`

5.41.1.4 `#define C3 (9.0/22.0)`

5.41.1.5 `#define C4 (3.0/26.0)`

5.41.1.6 `#define C5 (0.25*C3)`

5.41.1.7 `#define C6 (1.5*C4)`

5.41.1.8 `#define ERRRTOL 0.05`

5.41.1.9 `#define FMAX(a, b)`

Value:

```
(maxarg1=(a),maxarg2=(b),(maxarg1) > (maxarg2) ?\
(maxarg1) : (maxarg2))
```

5.41.1.10 `#define TINY 1.0e-25`

5.41.2 Function Documentation

5.41.2.1 `void Cycle_counter (int jf, int nsg, int c[])`

5.41.2.2 `void DC3D (float alpha, float X, float Y, float Z, float DEPTH, float DIP, float AL1, float AL2, float AW1, float AW2, float DISL1, float DISL2, float DISL3, float & UX, float & UY, float & UZ, float & UXX, float & UYX, float & UZX, float & UXY, float & UYX, float & UZY, float & UXZ, float & UYZ, float & UZZ, int & IRET, int CASE)`

5.41.2.3 `void gsl_eigenvalueproblem (int dimension, double eigenvalues[], double Matrix[])`

5.41.2.4 `float rd (float x, float y, float z)`

5.41.2.5 `void ref_func (double & x, double & y, double & z, double X1, double X2, double X3_plane, int flag_plane)`

5.41.2.6 `template<class type > void Rotazione_coord (type & x, type & y, int flag)`

5.41.2.7 `template<class type > void Rotazione_coord (type & x, type & y, double phi_degree)`

5.41.2.8 `template<class type > void Rotazione_tensore (type S[], double phi_degree)`

5.41.2.9 `template<class type > void Rotazione_vettore (type V[], double phi_degree)`

5.42 SRC/YANG.cpp File Reference

```
#include "YANG.h"
#include <cmath>
#include <iostream>
#include "PRINT_TEMPLATE.h"
```

Functions

- `template<typename T >`
`T sign (T t)`

5.42.1 Function Documentation

5.42.1.1 `template<typename T > T sign (T t)`

5.43 SRC/YANG.h File Reference

```
#include <string>
#include <cmath>
#include "PRESSURIZED.h"
```

Classes

- class [YANG](#)