

CRTomo  
multifreq\_dev

Generated by Doxygen 1.7.6.1

Mon Sep 16 2013 10:42:09



# Contents

<b>1</b>	<b>Data Type Index</b>	<b>1</b>
1.1	Data Types List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Type Documentation</b>	<b>7</b>
3.1	alloci Module Reference . . . . .	7
3.1.1	Member Data Documentation . . . . .	7
3.1.1.1	a . . . . .	7
3.1.1.2	adc . . . . .	7
3.1.1.3	ata . . . . .	7
3.1.1.4	ata_reg . . . . .	7
3.1.1.5	cov_d . . . . .	8
3.1.1.6	cov_m . . . . .	8
3.1.1.7	csens . . . . .	8
3.1.1.8	hpot . . . . .	8
3.1.1.9	hpotdc . . . . .	8
3.1.1.10	kpot . . . . .	8
3.1.1.11	kpotdc . . . . .	8
3.1.1.12	rnd_p . . . . .	8
3.1.1.13	rnd_r . . . . .	8
3.1.1.14	sens . . . . .	8
3.1.1.15	sensdc . . . . .	8
3.1.1.16	smatm . . . . .	8
3.2	bmcm_mod Module Reference . . . . .	8

3.2.1	Member Function/Subroutine Documentation	8
3.2.1.1	buncert	9
3.3	brough_mod Module Reference	9
3.3.1	Member Function/Subroutine Documentation	9
3.3.1.1	brough	9
3.4	bsmatm_mod Module Reference	9
3.4.1	Member Function/Subroutine Documentation	9
3.4.1.1	bsmatm	10
3.5	cg_mod Module Reference	10
3.5.1	Member Function/Subroutine Documentation	10
3.5.1.1	cjg	10
3.6	cjgmod Module Reference	10
3.6.1	Member Function/Subroutine Documentation	11
3.6.1.1	con_cjgmod	11
3.6.1.2	des_cjgmod	11
3.6.2	Member Data Documentation	11
3.6.2.1	ap	11
3.6.2.2	apdc	11
3.6.2.3	bvec	11
3.6.2.4	bvecdc	11
3.6.2.5	cgfac	11
3.6.2.6	cgres	11
3.6.2.7	cgres2	11
3.6.2.8	eps	11
3.6.2.9	ncg	11
3.6.2.10	ncgmax	11
3.6.2.11	pvec	12
3.6.2.12	pvecdc	12
3.6.2.13	rvec	12
3.6.2.14	rvecdc	12
3.7	datmod Module Reference	12
3.7.1	Member Data Documentation	13
3.7.1.1	iseed	13
3.7.1.2	kfak	13

3.7.1.3	<a href="#">lindiv</a>	13
3.7.1.4	<a href="#">Inse</a>	13
3.7.1.5	<a href="#">Inse2</a>	13
3.7.1.6	<a href="#">lpol</a>	13
3.7.1.7	<a href="#">lratio</a>	13
3.7.1.8	<a href="#">nanz</a>	13
3.7.1.9	<a href="#">npol</a>	13
3.7.1.10	<a href="#">nstabm0</a>	13
3.7.1.11	<a href="#">nstabp0</a>	13
3.7.1.12	<a href="#">nstabpA1</a>	13
3.7.1.13	<a href="#">nstabpA2</a>	13
3.7.1.14	<a href="#">nstabpB</a>	13
3.7.1.15	<a href="#">nstabw0</a>	13
3.7.1.16	<a href="#">sgmaa2</a>	13
3.7.1.17	<a href="#">sigmaa</a>	13
3.7.1.18	<a href="#">stabm0</a>	13
3.7.1.19	<a href="#">stabp0</a>	13
3.7.1.20	<a href="#">stabpA1</a>	13
3.7.1.21	<a href="#">stabpA2</a>	13
3.7.1.22	<a href="#">stabpB</a>	13
3.7.1.23	<a href="#">stabw0</a>	13
3.7.1.24	<a href="#">strnr</a>	14
3.7.1.25	<a href="#">strom</a>	14
3.7.1.26	<a href="#">vnr</a>	14
3.7.1.27	<a href="#">volt</a>	14
3.7.1.28	<a href="#">wmatd_cri</a>	14
3.7.1.29	<a href="#">wmatdp</a>	14
3.7.1.30	<a href="#">wmatdr</a>	14
3.8	<a href="#">electrmod Module Reference</a>	14
3.8.1	<a href="#">Member Data Documentation</a>	14
3.8.1.1	<a href="#">eanz</a>	14
3.8.1.2	<a href="#">enr</a>	14
3.9	<a href="#">elemmod Module Reference</a>	14
3.9.1	<a href="#">Detailed Description</a>	16

3.9.2	Member Data Documentation . . . . .	17
3.9.2.1	elanz . . . . .	17
3.9.2.2	elmam . . . . .	17
3.9.2.3	elmas . . . . .	17
3.9.2.4	elve . . . . .	17
3.9.2.5	esp_max . . . . .	17
3.9.2.6	esp_med . . . . .	17
3.9.2.7	esp_min . . . . .	17
3.9.2.8	esp_mit . . . . .	17
3.9.2.9	esp_std . . . . .	18
3.9.2.10	espx . . . . .	18
3.9.2.11	espy . . . . .	18
3.9.2.12	grid_max . . . . .	18
3.9.2.13	grid_maxx . . . . .	18
3.9.2.14	grid_maxy . . . . .	18
3.9.2.15	grid_min . . . . .	18
3.9.2.16	grid_minx . . . . .	18
3.9.2.17	grid_miny . . . . .	18
3.9.2.18	lrandb2 . . . . .	18
3.9.2.19	lsink . . . . .	19
3.9.2.20	mb . . . . .	19
3.9.2.21	nachbar . . . . .	19
3.9.2.22	nelanz . . . . .	19
3.9.2.23	nrel . . . . .	19
3.9.2.24	nsink . . . . .	19
3.9.2.25	relanz . . . . .	19
3.9.2.26	rnr . . . . .	19
3.9.2.27	sanz . . . . .	19
3.9.2.28	selanz . . . . .	19
3.9.2.29	smaxs . . . . .	20
3.9.2.30	snr . . . . .	20
3.9.2.31	sx . . . . .	20
3.9.2.32	sy . . . . .	20
3.9.2.33	sytopy . . . . .	20

3.9.2.34	typ	20
3.9.2.35	typanz	20
3.9.2.36	xk	20
3.9.2.37	yk	20
3.10	errmod Module Reference	21
3.10.1	Member Data Documentation	21
3.10.1.1	errflag	21
3.10.1.2	errnr	21
3.10.1.3	fetxt	21
3.10.1.4	fpcfg	21
3.10.1.5	fpcjg	21
3.10.1.6	fpeps	21
3.10.1.7	fperr	21
3.10.1.8	fpinv	21
3.10.1.9	fprun	21
3.11	femmod Module Reference	21
3.11.1	Member Data Documentation	22
3.11.1.1	b	22
3.11.1.2	bdc	22
3.11.1.3	elbg	22
3.11.1.4	fak	22
3.11.1.5	kg	22
3.11.1.6	lbeta	22
3.11.1.7	ldc	22
3.11.1.8	lrandb	22
3.11.1.9	lsr	22
3.11.1.10	pot	22
3.11.1.11	pota	22
3.11.1.12	relbg	22
3.12	get_ver Module Reference	22
3.12.1	Member Function/Subroutine Documentation	23
3.12.1.1	get_git_ver	23
3.12.2	Member Data Documentation	23
3.12.2.1	version	23

3.13	invhpmod Module Reference	23
3.13.1	Member Data Documentation	23
3.13.1.1	delectr	23
3.13.1.2	delem	23
3.13.1.3	dsens	24
3.13.1.4	dsigma	24
3.13.1.5	dstart	24
3.13.1.6	dstrom	24
3.13.1.7	dvolt	24
3.13.1.8	errnr2	24
3.13.1.9	izeit	24
3.13.1.10	j	24
3.13.1.11	k	24
3.13.1.12	kanal	24
3.13.1.13	l	24
3.13.1.14	lagain	24
3.13.1.15	lsetip	24
3.13.1.16	lsetup	24
3.13.1.17	tazeit	24
3.14	invmod Module Reference	24
3.14.1	Member Data Documentation	25
3.14.1.1	d0	25
3.14.1.2	dat	25
3.14.1.3	dpar	25
3.14.1.4	dpar2	25
3.14.1.5	fm0	25
3.14.1.6	lfpi	25
3.14.1.7	m0	25
3.14.1.8	m_ref	25
3.14.1.9	par	25
3.14.1.10	par_vari	25
3.14.1.11	wdfak	25
3.14.1.12	wmatd	25
3.14.1.13	wmatd2	25



---

3.14.1.14 wmfak . . . . .	25
3.15 konvmod Module Reference . . . . .	25
3.15.1 Member Data Documentation . . . . .	27
3.15.1.1 alam . . . . .	27
3.15.1.2 alfx . . . . .	27
3.15.1.3 alfz . . . . .	27
3.15.1.4 badmin . . . . .	27
3.15.1.5 bdpar . . . . .	27
3.15.1.6 betamgs . . . . .	27
3.15.1.7 betrms . . . . .	27
3.15.1.8 dlalt . . . . .	27
3.15.1.9 dlam . . . . .	27
3.15.1.10 fstart . . . . .	27
3.15.1.11 fstop . . . . .	27
3.15.1.12 it . . . . .	27
3.15.1.13 itmax . . . . .	27
3.15.1.14 itr . . . . .	28
3.15.1.15 l1min . . . . .	28
3.15.1.16 l1rat . . . . .	28
3.15.1.17 lam . . . . .	28
3.15.1.18 lam_cri . . . . .	28
3.15.1.19 lam_ref . . . . .	28
3.15.1.20 lam_ref_sw . . . . .	28
3.15.1.21 lamfix . . . . .	28
3.15.1.22 lammax . . . . .	28
3.15.1.23 lamnull_cri . . . . .	28
3.15.1.24 lamnull_fpi . . . . .	28
3.15.1.25 lcov1 . . . . .	28
3.15.1.26 lcov2 . . . . .	28
3.15.1.27 ldifff . . . . .	28
3.15.1.28 ldlamf . . . . .	28
3.15.1.29 ldlami . . . . .	28
3.15.1.30 lelerr . . . . .	28
3.15.1.31 lffhom . . . . .	28

---

3.15.1.32 lfphai . . . . .	28
3.15.1.33 lfstep . . . . .	28
3.15.1.34 lgauss . . . . .	28
3.15.1.35 llam . . . . .	28
3.15.1.36 llamf . . . . .	28
3.15.1.37 lsepri . . . . .	28
3.15.1.38 lphi0 . . . . .	29
3.15.1.39 lprior . . . . .	29
3.15.1.40 lres . . . . .	29
3.15.1.41 lrobust . . . . .	29
3.15.1.42 lsens . . . . .	29
3.15.1.43 lstep . . . . .	29
3.15.1.44 lsytop . . . . .	29
3.15.1.45 ltri . . . . .	29
3.15.1.46 lvario . . . . .	29
3.15.1.47 lverb . . . . .	29
3.15.1.48 lverb_dat . . . . .	29
3.15.1.49 lw_ref . . . . .	29
3.15.1.50 mqrms . . . . .	29
3.15.1.51 mswitch . . . . .	29
3.15.1.52 nlam . . . . .	29
3.15.1.53 nrmsd . . . . .	29
3.15.1.54 nrmsdm . . . . .	29
3.15.1.55 nx . . . . .	29
3.15.1.56 nz . . . . .	29
3.15.1.57 pharms . . . . .	29
3.15.1.58 rmsalt . . . . .	29
3.15.1.59 rmsreg . . . . .	29
3.15.1.60 rmssum . . . . .	29
3.15.1.61 rough . . . . .	29
3.15.1.62 step . . . . .	30
3.15.1.63 stpalt . . . . .	30
3.15.1.64 stpmin . . . . .	30
3.16 Make_noise Module Reference . . . . .	30

3.16.1	Member Function/Subroutine Documentation	30
3.16.1.1	get_noisemodel	30
3.16.1.2	Random_BreitWigner	30
3.16.1.3	Random_Exponential	31
3.16.1.4	Random_Gauss	31
3.16.1.5	Random_Init	31
3.16.1.6	write_noisemodel	31
3.17	modelmod Module Reference	31
3.17.1	Member Data Documentation	31
3.17.1.1	ind_ref_grad	31
3.17.1.2	manz	32
3.17.1.3	mnr	32
3.17.1.4	w_ref_im	32
3.17.1.5	w_ref_re	32
3.18	ompmo Module Reference	32
3.18.1	Member Data Documentation	32
3.18.1.1	CHUNK_0	32
3.18.1.2	CHUNK_1	32
3.18.1.3	CHUNK_2	32
3.18.1.4	CHUNK_3	32
3.18.1.5	NTHREADS	32
3.18.1.6	TID	32
3.19	pathmo Module Reference	32
3.19.1	Member Function/Subroutine Documentation	33
3.19.1.1	clear_string	33
3.19.2	Member Data Documentation	33
3.19.2.1	Inramd	33
3.19.2.2	mkdir	33
3.19.2.3	ramd	33
3.19.2.4	rmdir	33
3.19.2.5	slash	33
3.20	randbmo Module Reference	33
3.20.1	Member Data Documentation	34
3.20.1.1	rwd	34

3.20.1.2	<a href="#">rwdanz</a>	34
3.20.1.3	<a href="#">rwdbnr</a>	34
3.20.1.4	<a href="#">rwddc</a>	34
3.20.1.5	<a href="#">rwdnr</a>	34
3.20.1.6	<a href="#">rwn</a>	34
3.20.1.7	<a href="#">rwnanz</a>	34
3.20.1.8	<a href="#">rwndc</a>	34
3.21	<a href="#">sigmamod Module Reference</a>	34
3.21.1	<a href="#">Member Data Documentation</a>	34
3.21.1.1	<a href="#">bet0</a>	34
3.21.1.2	<a href="#">iseedpri</a>	34
3.21.1.3	<a href="#">lrho0</a>	34
3.21.1.4	<a href="#">lstart</a>	34
3.21.1.5	<a href="#">modl_stdn</a>	35
3.21.1.6	<a href="#">pha0</a>	35
3.21.1.7	<a href="#">sigma</a>	35
3.21.1.8	<a href="#">sigma0</a>	35
3.21.1.9	<a href="#">sigma2</a>	35
3.22	<a href="#">tic_toc Module Reference</a>	35
3.22.1	<a href="#">Member Function/Subroutine Documentation</a>	35
3.22.1.1	<a href="#">tic</a>	35
3.22.1.2	<a href="#">toc</a>	35
3.23	<a href="#">variomodel Module Reference</a>	35
3.23.1	<a href="#">Member Function/Subroutine Documentation</a>	36
3.23.1.1	<a href="#">get_vario</a>	36
3.23.1.2	<a href="#">mcova</a>	36
3.23.1.3	<a href="#">mvario</a>	36
3.23.1.4	<a href="#">set_vario</a>	36
3.24	<a href="#">wavenmod Module Reference</a>	36
3.24.1	<a href="#">Member Data Documentation</a>	36
3.24.1.1	<a href="#">amax</a>	36
3.24.1.2	<a href="#">amin</a>	36
3.24.1.3	<a href="#">kwn</a>	36
3.24.1.4	<a href="#">kwnanz</a>	36

3.24.1.5	kwnwi	36
3.24.1.6	swrtr	36
<b>4</b>	<b>File Documentation</b>	<b>39</b>
4.1	alloci.f90 File Reference	39
4.2	bbsedc.f90 File Reference	39
4.2.1	Function/Subroutine Documentation	39
4.2.1.1	bbsedc	39
4.3	bbsens.f90 File Reference	39
4.3.1	Function/Subroutine Documentation	39
4.3.1.1	bbsens	39
4.4	bsp_elem.f90 File Reference	40
4.4.1	Function/Subroutine Documentation	40
4.4.1.1	bsp_elem	40
4.5	bessi0.f90 File Reference	40
4.5.1	Function/Subroutine Documentation	41
4.5.1.1	BESSI0	41
4.6	bessi1.f90 File Reference	41
4.6.1	Function/Subroutine Documentation	41
4.6.1.1	BESSI1	41
4.7	bessk0.f90 File Reference	41
4.7.1	Function/Subroutine Documentation	41
4.7.1.1	BESSK0	41
4.8	bessk1.f90 File Reference	41
4.8.1	Function/Subroutine Documentation	41
4.8.1.1	BESSK1	41
4.9	beta.f90 File Reference	41
4.9.1	Function/Subroutine Documentation	42
4.9.1.1	beta	42
4.10	bkfak.f90 File Reference	42
4.10.1	Function/Subroutine Documentation	42
4.10.1.1	bkfak	42
4.11	blam0.f90 File Reference	42
4.11.1	Function/Subroutine Documentation	43

4.11.1.1	blam0	43
4.12	bmcm_mod.f90 File Reference	43
4.13	bnachbar.f90 File Reference	43
4.13.1	Function/Subroutine Documentation	43
4.13.1.1	bnachbar	44
4.14	bpar.f90 File Reference	44
4.14.1	Function/Subroutine Documentation	44
4.14.1.1	bpar	44
4.15	bpot.f90 File Reference	44
4.15.1	Function/Subroutine Documentation	45
4.15.1.1	bpot	45
4.16	brough_mod.f90 File Reference	45
4.17	bsendc.f90 File Reference	45
4.17.1	Function/Subroutine Documentation	45
4.17.1.1	bsendc	46
4.18	bsens.f90 File Reference	46
4.18.1	Function/Subroutine Documentation	46
4.18.1.1	bsens	46
4.19	bsensi.f90 File Reference	46
4.19.1	Function/Subroutine Documentation	47
4.19.1.1	bsensi	47
4.20	bsigm0.f90 File Reference	47
4.20.1	Function/Subroutine Documentation	47
4.20.1.1	bsigm0	47
4.21	bsigma.f90 File Reference	48
4.21.1	Function/Subroutine Documentation	48
4.21.1.1	bsigma	48
4.22	bsmatm_mod.f90 File Reference	48
4.23	bsytop.f90 File Reference	48
4.23.1	Function/Subroutine Documentation	49
4.23.1.1	bsytop	49
4.24	bvariogram.f90 File Reference	49
4.24.1	Function/Subroutine Documentation	49
4.24.1.1	bvariogram	49

4.25	bvariogram_s.f90 File Reference	50
4.25.1	Function/Subroutine Documentation	50
4.25.1.1	bvariogram_s	50
4.26	bvolt.f90 File Reference	50
4.26.1	Function/Subroutine Documentation	50
4.26.1.1	bvolt	51
4.27	bvolti.f90 File Reference	51
4.27.1	Function/Subroutine Documentation	51
4.27.1.1	bvolti	51
4.28	cg_mod.f90 File Reference	51
4.29	chareal.f90 File Reference	52
4.29.1	Function/Subroutine Documentation	52
4.29.1.1	chareal	52
4.30	chkpol.f90 File Reference	52
4.30.1	Function/Subroutine Documentation	52
4.30.1.1	chkpol	52
4.31	chol.f90 File Reference	52
4.31.1	Function/Subroutine Documentation	52
4.31.1.1	chol	53
4.32	chold.f90 File Reference	53
4.32.1	Function/Subroutine Documentation	53
4.32.1.1	chold	53
4.33	choldc.f90 File Reference	53
4.33.1	Function/Subroutine Documentation	53
4.33.1.1	choldc	54
4.34	cholz.f90 File Reference	54
4.34.1	Function/Subroutine Documentation	54
4.34.1.1	cholz	54
4.35	cjgmod.f90 File Reference	54
4.36	crerror.h File Reference	54
4.37	datmod.f90 File Reference	54
4.38	dmisft.f90 File Reference	55
4.38.1	Function/Subroutine Documentation	55
4.38.1.1	chkpo2	55

4.38.1.2 dmsft . . . . .	55
4.39 electrmod.f90 File Reference . . . . .	56
4.40 elem1.f90 File Reference . . . . .	56
4.40.1 Function/Subroutine Documentation . . . . .	56
4.40.1.1 elem1 . . . . .	56
4.41 elem3.f90 File Reference . . . . .	57
4.41.1 Function/Subroutine Documentation . . . . .	57
4.41.1.1 elem3 . . . . .	57
4.42 elem4.f90 File Reference . . . . .	57
4.42.1 Function/Subroutine Documentation . . . . .	57
4.42.1.1 elem4 . . . . .	57
4.43 elem5.f90 File Reference . . . . .	57
4.43.1 Function/Subroutine Documentation . . . . .	58
4.43.1.1 elem5 . . . . .	58
4.44 elem8.f90 File Reference . . . . .	58
4.44.1 Function/Subroutine Documentation . . . . .	58
4.44.1.1 elem8 . . . . .	58
4.45 elemmod.f90 File Reference . . . . .	59
4.46 errmod.f90 File Reference . . . . .	59
4.47 fem.f90 File Reference . . . . .	59
4.47.1 Function/Subroutine Documentation . . . . .	59
4.47.1.1 fem . . . . .	60
4.48 femmod.f90 File Reference . . . . .	61
4.49 filpat.f90 File Reference . . . . .	61
4.49.1 Function/Subroutine Documentation . . . . .	61
4.49.1.1 filpat . . . . .	61
4.50 gammln.f90 File Reference . . . . .	61
4.50.1 Function/Subroutine Documentation . . . . .	61
4.50.1.1 gammln . . . . .	61
4.51 gaulag.f90 File Reference . . . . .	61
4.51.1 Function/Subroutine Documentation . . . . .	61
4.51.1.1 gaulag . . . . .	62
4.52 gauleg.f90 File Reference . . . . .	62
4.52.1 Function/Subroutine Documentation . . . . .	62



---

4.52.1.1	gauleg	62
4.53	gauss_cmplx.f90 File Reference	63
4.53.1	Function/Subroutine Documentation	63
4.53.1.1	Gauss_cmplx	63
4.54	gauss_dble.f90 File Reference	63
4.54.1	Function/Subroutine Documentation	63
4.54.1.1	Gauss_dble	63
4.55	get_error.f90 File Reference	63
4.55.1	Function/Subroutine Documentation	63
4.55.1.1	get_error	64
4.56	get_git_ver.f90 File Reference	64
4.57	get_unit.f90 File Reference	64
4.57.1	Function/Subroutine Documentation	64
4.57.1.1	get_unit	65
4.57.1.2	read_comments	66
4.58	intcha.f90 File Reference	66
4.59	inv.f90 File Reference	66
4.59.1	Function/Subroutine Documentation	66
4.59.1.1	inv	67
4.60	invhpmmod.f90 File Reference	68
4.61	invmod.f90 File Reference	68
4.62	kompab.f90 File Reference	68
4.62.1	Function/Subroutine Documentation	68
4.62.1.1	kompab	68
4.63	kompadc.f90 File Reference	68
4.63.1	Function/Subroutine Documentation	69
4.63.1.1	kompadc	69
4.64	kompb.f90 File Reference	69
4.64.1	Function/Subroutine Documentation	69
4.64.1.1	kompb	70
4.65	kompbdc.f90 File Reference	70
4.65.1	Function/Subroutine Documentation	70
4.65.1.1	kompbdc	70
4.66	kont1.f90 File Reference	70

---

4.66.1	Function/Subroutine Documentation	71
4.66.1.1	kont1	71
4.67	kont2.f90 File Reference	71
4.67.1	Function/Subroutine Documentation	71
4.67.1.1	kont2	71
4.68	konvmod.f90 File Reference	72
4.69	linvd.f90 File Reference	72
4.69.1	Function/Subroutine Documentation	72
4.69.1.1	linvd	72
4.70	linvz.f90 File Reference	72
4.70.1	Function/Subroutine Documentation	72
4.70.1.1	linvz	72
4.71	make_noise.f90 File Reference	72
4.72	mdian1.f90 File Reference	72
4.72.1	Function/Subroutine Documentation	73
4.72.1.1	MDIAN1	73
4.73	modelmod.f90 File Reference	73
4.74	my_git_version.h File Reference	73
4.75	ompmmod.f90 File Reference	73
4.76	parfit.f90 File Reference	74
4.76.1	Function/Subroutine Documentation	74
4.76.1.1	parfit	74
4.77	pathmod.f90 File Reference	74
4.78	potana.f90 File Reference	74
4.78.1	Function/Subroutine Documentation	74
4.78.1.1	potana	75
4.79	precal.f90 File Reference	75
4.79.1	Function/Subroutine Documentation	75
4.79.1.1	precal	76
4.80	rall.f90 File Reference	76
4.80.1	Function/Subroutine Documentation	77
4.80.1.1	rall	77
4.81	randb.f90 File Reference	78
4.81.1	Function/Subroutine Documentation	78

4.81.1.1	randb	78
4.82	randb2.f90 File Reference	78
4.82.1	Function/Subroutine Documentation	79
4.82.1.1	randb2	79
4.83	randbdc2.f90 File Reference	79
4.83.1	Function/Subroutine Documentation	79
4.83.1.1	randbdc2	79
4.84	randbmod.f90 File Reference	80
4.85	randdc.f90 File Reference	80
4.85.1	Function/Subroutine Documentation	80
4.85.1.1	randdc	80
4.86	rdati.f90 File Reference	80
4.86.1	Function/Subroutine Documentation	80
4.86.1.1	rdati	81
4.87	rdatm.f90 File Reference	81
4.87.1	Function/Subroutine Documentation	81
4.87.1.1	rdatm	82
4.88	refsig.f90 File Reference	82
4.88.1	Function/Subroutine Documentation	82
4.88.1.1	refsig	82
4.89	relectr.f90 File Reference	82
4.89.1	Function/Subroutine Documentation	83
4.89.1.1	relectr	83
4.90	relem.f90 File Reference	83
4.90.1	Function/Subroutine Documentation	83
4.90.1.1	relem	84
4.91	rrandb.f90 File Reference	84
4.91.1	Function/Subroutine Documentation	84
4.91.1.1	rrandb	84
4.92	rsigma.f90 File Reference	84
4.92.1	Function/Subroutine Documentation	85
4.92.1.1	rsigma	85
4.92.1.2	set_ind_ref_grad	85
4.92.1.3	set_ind_ref_grad2	86

4.93 rtrafo.f90 File Reference . . . . .	86
4.93.1 Function/Subroutine Documentation . . . . .	86
4.93.1.1 rtrafo . . . . .	86
4.94 rwaven.f90 File Reference . . . . .	86
4.94.1 Function/Subroutine Documentation . . . . .	87
4.94.1.1 rwaven . . . . .	87
4.95 scalab.f90 File Reference . . . . .	87
4.95.1 Function/Subroutine Documentation . . . . .	87
4.95.1.1 scalab . . . . .	88
4.96 scaldc.f90 File Reference . . . . .	88
4.96.1 Function/Subroutine Documentation . . . . .	88
4.96.1.1 scaldc . . . . .	88
4.97 semi-variogram.f90 File Reference . . . . .	88
4.97.1 Function/Subroutine Documentation . . . . .	89
4.97.1.1 semi_variogram . . . . .	89
4.98 sigmamod.f90 File Reference . . . . .	89
4.99 sort.f90 File Reference . . . . .	89
4.99.1 Function/Subroutine Documentation . . . . .	89
4.99.1.1 SORT . . . . .	89
4.100tic_toc.f90 File Reference . . . . .	89
4.101update.f90 File Reference . . . . .	90
4.101.1 Function/Subroutine Documentation . . . . .	90
4.101.1.1 update . . . . .	90
4.102variomodel.f90 File Reference . . . . .	91
4.103vre.f90 File Reference . . . . .	91
4.103.1 Function/Subroutine Documentation . . . . .	91
4.103.1.1 vre . . . . .	91
4.104vredc.f90 File Reference . . . . .	91
4.104.1 Function/Subroutine Documentation . . . . .	91
4.104.1.1 vredc . . . . .	92
4.105wavenmod.f90 File Reference . . . . .	92
4.106wdatm.f90 File Reference . . . . .	92
4.106.1 Function/Subroutine Documentation . . . . .	92
4.106.1.1 wdatm . . . . .	92

---

4.107wkpot.f90 File Reference . . . . .	93
4.107.1 Function/Subroutine Documentation . . . . .	93
4.107.1.1 wkpot . . . . .	93
4.108wout.f90 File Reference . . . . .	93
4.108.1 Function/Subroutine Documentation . . . . .	93
4.108.1.1 wout . . . . .	93
4.109wout_up.f90 File Reference . . . . .	94
4.109.1 Function/Subroutine Documentation . . . . .	94
4.109.1.1 wout_up . . . . .	94
4.110wpot.f90 File Reference . . . . .	94
4.110.1 Function/Subroutine Documentation . . . . .	94
4.110.1.1 wpot . . . . .	95
4.111wsens.f90 File Reference . . . . .	95
4.111.1 Function/Subroutine Documentation . . . . .	95
4.111.1.1 wsens . . . . .	96



# Chapter 1

## Data Type Index

### 1.1 Data Types List

Here are the data types with brief descriptions:

<a href="#">alloci</a>	7
<a href="#">bmcm_mod</a>	8
<a href="#">brough_mod</a>	9
<a href="#">bsmatm_mod</a>	9
<a href="#">cg_mod</a>	10
<a href="#">cjgmod</a>	10
<a href="#">datmod</a>	12
<a href="#">electrmod</a>	14
<a href="#">elemmod</a>	
Replacement of former 'elem.fin' and basically contains the FE- element related variables and two methods for allocation and deallo- cation of global memory	14
<a href="#">errmod</a>	21
<a href="#">femmod</a>	21
<a href="#">get_ver</a>	22
<a href="#">invhpmode</a>	23
<a href="#">invmod</a>	24
<a href="#">konvmod</a>	25
<a href="#">Make_noise</a>	30
<a href="#">modelmod</a>	31
<a href="#">ompmode</a>	32
<a href="#">pathmod</a>	32
<a href="#">randbmod</a>	33
<a href="#">sigmamod</a>	34
<a href="#">tic_toc</a>	35
<a href="#">variomodel</a>	35
<a href="#">wavenmod</a>	36





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">alloci.f90</a>	39
<a href="#">bbsedc.f90</a>	39
<a href="#">bbsens.f90</a>	39
<a href="#">besp_elem.f90</a>	40
<a href="#">bessi0.f90</a>	40
<a href="#">bessi1.f90</a>	41
<a href="#">bessk0.f90</a>	41
<a href="#">bessk1.f90</a>	41
<a href="#">beta.f90</a>	41
<a href="#">bkfak.f90</a>	42
<a href="#">blam0.f90</a>	42
<a href="#">bmcm_mod.f90</a>	43
<a href="#">bnachbar.f90</a>	43
<a href="#">bpar.f90</a>	44
<a href="#">bpot.f90</a>	44
<a href="#">brough_mod.f90</a>	45
<a href="#">bsendc.f90</a>	45
<a href="#">bsens.f90</a>	46
<a href="#">bsensi.f90</a>	46
<a href="#">bsigm0.f90</a>	47
<a href="#">bsigma.f90</a>	48
<a href="#">bsmatm_mod.f90</a>	48
<a href="#">bsytop.f90</a>	48
<a href="#">bvariogram.f90</a>	49
<a href="#">bvariogram_s.f90</a>	50
<a href="#">bvolt.f90</a>	50
<a href="#">bvolti.f90</a>	51
<a href="#">cg_mod.f90</a>	51
<a href="#">chareal.f90</a>	52

chkpol.f90	52
chol.f90	52
chold.f90	53
choldc.f90	53
cholz.f90	54
cjgmod.f90	54
crerror.h	54
datmod.f90	54
dmisft.f90	55
electrmod.f90	56
elem1.f90	56
elem3.f90	57
elem4.f90	57
elem5.f90	57
elem8.f90	58
elemmod.f90	59
errmod.f90	59
fem.f90	59
femmod.f90	61
filpat.f90	61
gammln.f90	61
gaulag.f90	61
gauleg.f90	62
gauss_cmplx.f90	63
gauss_dble.f90	63
get_error.f90	63
get_git_ver.f90	64
get_unit.f90	64
intcha.f90	66
inv.f90	66
invhpmmod.f90	68
invmod.f90	68
kompab.f90	68
kompadc.f90	68
kompb.f90	69
kompbdc.f90	70
kont1.f90	70
kont2.f90	71
konvmod.f90	72
linvd.f90	72
linvz.f90	72
make_noise.f90	72
mdian1.f90	72
modelmod.f90	73
my_git_version.h	73
ompmmod.f90	73
parfit.f90	74
pathmod.f90	74
potana.f90	74
precal.f90	75

rall.f90	76
randb.f90	78
randb2.f90	78
randbdc2.f90	79
randbmod.f90	80
randdc.f90	80
rdati.f90	80
rdatm.f90	81
refsig.f90	82
relectr.f90	82
relem.f90	83
rrandb.f90	84
rsigma.f90	84
rtrafo.f90	86
rwaven.f90	86
scalab.f90	87
scaldc.f90	88
semi-variogram.f90	88
sigmamod.f90	89
sort.f90	89
tic_toc.f90	89
update.f90	90
variomodel.f90	91
vre.f90	91
vredc.f90	91
wavenmod.f90	92
wdatm.f90	92
wkpot.f90	93
wout.f90	93
wout_up.f90	94
wpot.f90	94
wsens.f90	95



## Chapter 3

# Data Type Documentation

### 3.1 alloci Module Reference

#### Public Attributes

- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [a](#)
- COMPLEX(KIND(0D0)), dimension(:, :, :), allocatable, public [kpot](#)
- COMPLEX(KIND(0D0)), dimension(:, :), allocatable, public [hpot](#)
- COMPLEX(KIND(0D0)), dimension(:, :), allocatable, public [sens](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [csens](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [adc](#)
- REAL(KIND(0D0)), dimension(:, :, :), allocatable, public [kpotdc](#)
- REAL(KIND(0D0)), dimension(:, :), allocatable, public [hpotdc](#)
- REAL(KIND(0D0)), dimension(:, :), allocatable, public [sensdc](#)
- REAL(KIND(0D0)), dimension(:, :), allocatable, public [cov\\_d](#)
- REAL(KIND(0D0)), dimension(:, :), allocatable, public [smatm](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [rnd\\_r](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [rnd\\_p](#)
- REAL(KIND(0D0)), dimension(:, :), allocatable, public [ata](#)
- REAL(KIND(0D0)), dimension(:, :), allocatable, public [ata\\_reg](#)
- REAL(KIND(0D0)), dimension(:, :), allocatable, public [cov\\_m](#)

#### 3.1.1 Member Data Documentation

3.1.1.1 COMPLEX (KIND(0D0)), dimension(:), allocatable, public alloci::a

3.1.1.2 REAL (KIND(0D0)), dimension(:), allocatable, public alloci::adc

3.1.1.3 REAL (KIND(0D0)), dimension(:, :), allocatable, public alloci::ata

3.1.1.4 REAL (KIND(0D0)), dimension(:, :), allocatable, public alloci::ata\_reg

- 3.1.1.5 REAL (KIND(0D0)), dimension(:, :), allocatable, public alloci::cov\_d
- 3.1.1.6 REAL (KIND(0D0)), dimension(:, :), allocatable, public alloci::cov\_m
- 3.1.1.7 REAL (KIND(0D0)), dimension(:), allocatable, public alloci::csens
- 3.1.1.8 COMPLEX (KIND(0D0)), dimension(:, :), allocatable, public alloci::hpot
- 3.1.1.9 REAL (KIND(0D0)), dimension(:, :), allocatable, public alloci::hpotdc
- 3.1.1.10 COMPLEX (KIND(0D0)), dimension(:, :, :), allocatable, public alloci::kpot
- 3.1.1.11 REAL (KIND(0D0)), dimension(:, :, :), allocatable, public alloci::kpotdc
- 3.1.1.12 REAL (KIND(0D0)), dimension(:), allocatable, public alloci::rnd\_p
- 3.1.1.13 REAL (KIND(0D0)), dimension(:), allocatable, public alloci::rnd\_r
- 3.1.1.14 COMPLEX (KIND(0D0)), dimension(:, :), allocatable, public alloci::sens
- 3.1.1.15 REAL (KIND(0D0)), dimension(:, :), allocatable, public alloci::sensdc
- 3.1.1.16 REAL (KIND(0D0)), dimension(:, :), allocatable, public alloci::smatm

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

- [alloci.f90](#)

## 3.2 bmcm\_mod Module Reference

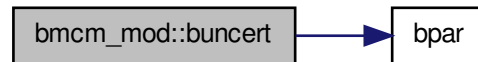
### Public Member Functions

- subroutine, public [buncert](#) (kanal, lamalt)

### 3.2.1 Member Function/Subroutine Documentation

3.2.1.1 subroutine, public **bmcm\_mod::buncert** ( INTEGER (KIND = 4 ), intent(in) *kanal*,  
REAL (KIND(OD0)), intent(in) *lamalt* )

Here is the call graph for this function:



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

- [bmcm\\_mod.f90](#)

### 3.3 brough\_mod Module Reference

#### Public Member Functions

- subroutine, public [brough](#)

#### 3.3.1 Member Function/Subroutine Documentation

3.3.1.1 subroutine, public **brough\_mod::brough** ( )

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

- [brough\\_mod.f90](#)

### 3.4 bsmatm\_mod Module Reference

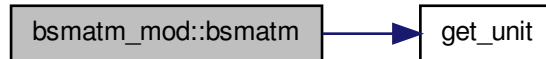
#### Public Member Functions

- subroutine, public [bsmatm](#) (it, l\_bsmat)

#### 3.4.1 Member Function/Subroutine Documentation

3.4.1.1 subroutine, public **bsmatm\_mod::bsmatm** ( INTEGER (KIND = 4 ), intent(in) *it*, LOGICAL, intent(inout) *l\_bsmat* )

Here is the call graph for this function:



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

- [bsmatm\\_mod.f90](#)

## 3.5 cg\_mod Module Reference

### Public Member Functions

- subroutine, public [cjb](#)  
*controls whather we have REAL or COMPLEX case*

### 3.5.1 Member Function/Subroutine Documentation

3.5.1.1 subroutine, public **cg\_mod::cjb** ( )

controls whather we have REAL or COMPLEX case

Subroutine calculates model update (DC) with preconditioned conjugate gradient method cjb flow control subroutine is called from outside and checks for the different cases (DC,IP,FPI)

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

- [cg\\_mod.f90](#)

## 3.6 cjbmod Module Reference

### Public Member Functions

- subroutine, public [con\\_cjbmod](#) (mycase, errtxt, errnr)
- subroutine, public [des\\_cjbmod](#) (mycase, errtxt, errnr)



**Public Attributes**

- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [ap](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [bvec](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [rvec](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [pvec](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [apdc](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [bvecdc](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [rvecdc](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [pvecdc](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [cgres](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [cgres2](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [cgfac](#)
- REAL(KIND(0D0)), public [eps](#)
- INTEGER(KIND=4), public [ncgmax](#)
- INTEGER(KIND=4), public [ncg](#)

**3.6.1 Member Function/Subroutine Documentation**

- 3.6.1.1 subroutine, public **cjgmod::con\_cjgmod** ( INTEGER (KIND=4), intent(in) *mycase*, CHARACTER (\*), intent(inout) *errtxt*, INTEGER (KIND=4), intent(out) *errnr* )
- 3.6.1.2 subroutine, public **cjgmod::des\_cjgmod** ( INTEGER (KIND=4), intent(in) *mycase*, CHARACTER (\*), intent(inout) *errtxt*, INTEGER (KIND=4), intent(out) *errnr* )

**3.6.2 Member Data Documentation**

- 3.6.2.1 COMPLEX(KIND(0D0)), dimension(:), allocatable, public **cjgmod::ap**
- 3.6.2.2 REAL(KIND(0D0)), dimension(:), allocatable, public **cjgmod::apdc**
- 3.6.2.3 COMPLEX(KIND(0D0)), dimension(:), allocatable, public **cjgmod::bvec**
- 3.6.2.4 REAL(KIND(0D0)), dimension(:), allocatable, public **cjgmod::bvecdc**
- 3.6.2.5 REAL(KIND(0D0)), dimension(:), allocatable, public **cjgmod::cgfac**
- 3.6.2.6 REAL(KIND(0D0)), dimension(:), allocatable, public **cjgmod::cgres**
- 3.6.2.7 REAL(KIND(0D0)), dimension(:), allocatable, public **cjgmod::cgres2**
- 3.6.2.8 REAL(KIND(0D0)), public **cjgmod::eps**
- 3.6.2.9 INTEGER (KIND = 4), public **cjgmod::ncg**
- 3.6.2.10 INTEGER (KIND = 4), public **cjgmod::ncgmax**

3.6.2.11 `COMPLEX(KIND(0D0))`, `dimension(:)`, allocatable, public `cjgmod::pvec`

3.6.2.12 `REAL(KIND(0D0))`, `dimension(:)`, allocatable, public `cjgmod::pvecdc`

3.6.2.13 `COMPLEX(KIND(0D0))`, `dimension(:)`, allocatable, public `cjgmod::rvec`

3.6.2.14 `REAL(KIND(0D0))`, `dimension(:)`, allocatable, public `cjgmod::rvecdc`

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

- [cjgmod.f90](#)

## 3.7 datmod Module Reference

### Public Attributes

- `INTEGER(KIND=4)`, public [nanz](#)
- `INTEGER(KIND=4)`, `dimension(:)`, allocatable, public [strnr](#)
- `REAL(KIND(0D0))`, `dimension(:)`, allocatable, public [strom](#)
- `INTEGER(KIND=4)`, `dimension(:)`, allocatable, public [vnr](#)
- `COMPLEX(KIND(0D0))`, `dimension(:)`, allocatable, public [volt](#)
- `COMPLEX(KIND(0D0))`, `dimension(:)`, allocatable, public [sigmaa](#)
- `COMPLEX(KIND(0D0))`, `dimension(:)`, allocatable, public [sgmaa2](#)
- `REAL(KIND(0D0))`, `dimension(:)`, allocatable, public [kfak](#)
- `REAL(KIND(0D0))`, `dimension(:)`, allocatable, public [wmatdp](#)
- `REAL(KIND(0D0))`, `dimension(:)`, allocatable, public [wmatdr](#)
- `REAL(KIND(0D0))`, `dimension(:)`, allocatable, public [wmatd\\_cri](#)
- `REAL(KIND(0D0))`, public [stabw0](#)
- `REAL(KIND(0D0))`, public [stabm0](#)
- `REAL(KIND(0D0))`, public [stabp0](#)
- `REAL(KIND(0D0))`, public [stabpA1](#)
- `REAL(KIND(0D0))`, public [stabpB](#)
- `REAL(KIND(0D0))`, public [stabpA2](#)
- `LOGICAL`, public [lindiv](#)
- `LOGICAL`, public [lratio](#)
- `LOGICAL`, public [lpol](#)
- `LOGICAL`, public [lase](#)
- `LOGICAL`, public [lase2](#)
- `INTEGER(KIND=4)`, public [iseed](#)
- `REAL(KIND(0D0))`, public [nstabw0](#)
- `REAL(KIND(0D0))`, public [nstabm0](#)
- `REAL(KIND(0D0))`, public [nstabpB](#)
- `REAL(KIND(0D0))`, public [nstabpA1](#)
- `REAL(KIND(0D0))`, public [nstabpA2](#)
- `REAL(KIND(0D0))`, public [nstabp0](#)
- `INTEGER(KIND=4)`, public [npol](#)

### 3.7.1 Member Data Documentation

- 3.7.1.1 INTEGER (KIND = 4), public datmod::iseed
- 3.7.1.2 REAL (KIND(0D0)), dimension(:), allocatable, public datmod::kfak
- 3.7.1.3 LOGICAL, public datmod::lindiv
- 3.7.1.4 LOGICAL, public datmod::lnse
- 3.7.1.5 LOGICAL, public datmod::lnse2
- 3.7.1.6 LOGICAL, public datmod::lpol
- 3.7.1.7 LOGICAL, public datmod::lratio
- 3.7.1.8 INTEGER (KIND = 4), public datmod::nanz
- 3.7.1.9 INTEGER (KIND = 4), public datmod::npol
- 3.7.1.10 REAL (KIND(0D0)), public datmod::nstabm0
- 3.7.1.11 REAL (KIND(0D0)), public datmod::nstabp0
- 3.7.1.12 REAL (KIND(0D0)), public datmod::nstabpA1
- 3.7.1.13 REAL (KIND(0D0)), public datmod::nstabpA2
- 3.7.1.14 REAL (KIND(0D0)), public datmod::nstabpB
- 3.7.1.15 REAL (KIND(0D0)), public datmod::nstabw0
- 3.7.1.16 COMPLEX (KIND(0D0)), dimension(:), allocatable, public datmod::sgmaa2
- 3.7.1.17 COMPLEX (KIND(0D0)), dimension(:), allocatable, public datmod::sigmaa
- 3.7.1.18 REAL (KIND(0D0)), public datmod::stabm0
- 3.7.1.19 REAL (KIND(0D0)), public datmod::stabp0
- 3.7.1.20 REAL (KIND(0D0)), public datmod::stabpA1
- 3.7.1.21 REAL (KIND(0D0)), public datmod::stabpA2
- 3.7.1.22 REAL (KIND(0D0)), public datmod::stabpB
- 3.7.1.23 REAL (KIND(0D0)), public datmod::stabw0

3.7.1.24 INTEGER (KIND = 4), dimension(:), allocatable, public `datmod::strnr`

3.7.1.25 REAL (KIND(0D0)), dimension(:), allocatable, public `datmod::strom`

3.7.1.26 INTEGER (KIND = 4), dimension(:), allocatable, public `datmod::vnr`

3.7.1.27 COMPLEX (KIND(0D0)), dimension(:), allocatable, public `datmod::volt`

3.7.1.28 REAL (KIND(0D0)), dimension(:), allocatable, public `datmod::wmatd_cri`

3.7.1.29 REAL (KIND(0D0)), dimension(:), allocatable, public `datmod::wmatdp`

3.7.1.30 REAL (KIND(0D0)), dimension(:), allocatable, public `datmod::wmatdr`

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

- [datmod.f90](#)

## 3.8 electrmod Module Reference

### Public Attributes

- INTEGER(KIND=4), public [eanz](#)
- INTEGER(KIND=4), dimension(:), allocatable, public [enr](#)

### 3.8.1 Member Data Documentation

3.8.1.1 INTEGER(KIND = 4), public `electrmod::eanz`

3.8.1.2 INTEGER(KIND = 4), dimension(:), allocatable, public `electrmod::enr`

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

- [electrmod.f90](#)

## 3.9 elemmod Module Reference

Replacement of former 'elem.fin' and basically contains the FE-element related variables and two methods for allocation and deallocation of global memory.

### Public Attributes

- INTEGER(KIND=4), public [sanz](#)

- Anzahl der Knoten (bzw. Knotenvariablen)*
- INTEGER(KIND=4), public [typanz](#)
- Anzahl der Elementtypen.*
- INTEGER(KIND=4), public [mb](#)
- Bandbreite der Gesamtsteifigkeitsmatrix 'a'.*
- INTEGER(KIND=4), dimension(:), allocatable, public [typ](#)
- Elementtypen (Randelemente (ntyp > 10) am Schluss !)*
- INTEGER(KIND=4), dimension(:), allocatable, public [nelanz](#)
- Anzahl der Elemente eines bestimmten Typs.*
- INTEGER(KIND=4), dimension(:), allocatable, public [selanz](#)
- Anzahl der Knoten (bzw. Knotenvariablen) in einem Elementtyp.*
- INTEGER(KIND=4), dimension(:), allocatable, public [snr](#)
- Zeiger auf Koordinaten der Knoten (Inverser Permutationsvektor der Umnummerierung)*
- REAL(KIND(0D0)), dimension(:), allocatable, public [sx](#)
- x-Koordinaten der Knoten*
- REAL(KIND(0D0)), dimension(:), allocatable, public [sy](#)
- y-Koordinaten der Knoten*
- REAL(KIND(0D0)), dimension(:), allocatable, public [espx](#)
- Elementschwerpunktkoordinaten (ESP) der Flaechelemente x-direction.*
- REAL(KIND(0D0)), dimension(:), allocatable, public [espy](#)
- Elementschwerpunktkoordinaten (ESP) der Flaechelemente y-direction.*
- INTEGER, dimension(:,:), allocatable, public [nachbar](#)
- Zeiger auf die Nachbarn der nichtentarteten Elemente.*
- INTEGER(KIND=4), dimension(:,:), allocatable, public [nrel](#)
- Knotennummern der Elemente (Reihenfolge !)*
- INTEGER(KIND=4), public [elanz](#)
- Anzahl der Elemente (ohne Randelemente)*
- INTEGER(KIND=4), public [relanz](#)
- Anzahl der Randelemente.*
- INTEGER(KIND=4), dimension(:), allocatable, public [rnr](#)
- Zeiger auf Werte der Randelemente.*
- INTEGER(KIND=4), public [smaxs](#)
- Groeste Anzahl der Knoten der Flaechelemente.*
- REAL(KIND(0D0)), public [esp\\_min](#)
- Gitter statistiken: Minimaler Abstand zwischen (Flaechen) Elementschwerpunkten.*
- REAL(KIND(0D0)), public [esp\\_max](#)
- Gitter statistiken: Maximaler Abstand zwischen (Flaechen) Elementschwerpunkten.*
- REAL(KIND(0D0)), public [esp\\_mit](#)
- Gitter statistiken: Mittelwert/Median und Standardabweichung der ESP.*
- REAL(KIND(0D0)), public [esp\\_med](#)
- Gitter statistiken: Median und Standardabweichung der ESP.*
- REAL(KIND(0D0)), public [esp\\_std](#)
- Gitter statistiken: Standardabweichung der ESP.*

- REAL(KIND(0D0)), public [grid\\_min](#)  
*Minaler Gitterabstand (Betrag)*
- REAL(KIND(0D0)), public [grid\\_max](#)  
*Maximaler Gitterabstand (Betrag)*
- REAL(KIND(0D0)), public [grid\\_minx](#)  
*Minimaler Gitterabstand in x-Richtung.*
- REAL(KIND(0D0)), public [grid\\_miny](#)  
*Minimaler Gitterabstand in y-Richtung.*
- REAL(KIND(0D0)), public [grid\\_maxx](#)  
*Maximaler Gitterabstand in x-Richtung.*
- REAL(KIND(0D0)), public [grid\\_maxy](#)  
*Maximaler Gitterabstand in y-Richtung.*
- LOGICAL, public [lsink](#)  
*switch/number fictitious sink node (only for 2D)*
- INTEGER(KIND=4), public [nsink](#)  
*number of grid node for sink*
- LOGICAL, public [lrandb2](#)  
*switch boundary values*
- REAL(KIND(0D0)), public [sytop](#)  
*mittlere y-Koordinate aller Randelemente vom Typ 12 ("no flow")*
- REAL(KIND(0D0)), dimension(:), allocatable, public [xk](#)  
*x-Koordinaten der Eckknotenpunkte*
- REAL(KIND(0D0)), dimension(:), allocatable, public [yk](#)  
*y-Koordinaten der Eckknotenpunkte*
- REAL(KIND(0D0)), dimension(:,,:), allocatable, public [elmam](#)  
*Elementarmatrizen.*
- REAL(KIND(0D0)), dimension(:,,:), allocatable, public [elmas](#)  
*Elementarmatrizen.*
- REAL(KIND(0D0)), dimension(:), allocatable, public [elve](#)  
*Elementvektor.*

### 3.9.1 Detailed Description

Replacement of former 'elem.fin' and basically contains the FE-element related variables and two methods for allocation and deallocation of global memory.

#### Author

Andreas Kemna

- 24-Nov-1993, elem.fin was written

Roland Martin

- 20-Nov-2009 until Sep. 2013
- translated to F90 module

- added nachbar (neighbor)
- added esp (central point)
- added some variables associated with grid statistics
- add doxy style for comments
- add description and translation

### 3.9.2 Member Data Documentation

#### 3.9.2.1 INTEGER(KIND = 4), public elemmod::elanz

Anzahl der Elemente (ohne Randelemente)

#### 3.9.2.2 REAL(KIND(0D0)), dimension(:, :), allocatable, public elemmod::elmam

Elementarmatrizen.

#### 3.9.2.3 REAL(KIND(0D0)), dimension(:, :), allocatable, public elemmod::elmas

Elementarmatrizen.

#### 3.9.2.4 REAL(KIND(0D0)), dimension(:), allocatable, public elemmod::elve

Elementvektor.

#### 3.9.2.5 REAL(KIND(0D0)), public elemmod::esp\_max

Gitter statistiken: Maximaler Abstand zwischen (Flaechen) Elementschwerpunkten.

#### 3.9.2.6 REAL(KIND(0D0)), public elemmod::esp\_med

Gitter statistiken: Median und Standardabweichung der ESP.

#### 3.9.2.7 REAL(KIND(0D0)), public elemmod::esp\_min

Gitter statistiken: Minimaler Abstand zwischen (Flaechen) Elementschwerpunkten.

#### 3.9.2.8 REAL(KIND(0D0)), public elemmod::esp\_mit

Gitter statistiken: Mittelwert/Median und Standardabweichung der ESP.

**3.9.2.9 REAL(KIND(0D0)), public elemmod::esp\_std**

Gitter statistiken: Standardabweichung der ESP.

**3.9.2.10 REAL(KIND(0D0)), dimension(:), allocatable, public elemmod::espx**

Elementschwerpunktkoordinaten (ESP) der Flaechelemente x-direction.

**3.9.2.11 REAL(KIND(0D0)), dimension(:), allocatable, public elemmod::espy**

Elementschwerpunktkoordinaten (ESP) der Flaechelemente y-direction.

**3.9.2.12 REAL(KIND(0D0)), public elemmod::grid\_max**

Maximaler Gitterabstand (Betrag)

**3.9.2.13 REAL(KIND(0D0)), public elemmod::grid\_maxx**

Maximaler Gitterabstand in x-Richtung.

**3.9.2.14 REAL(KIND(0D0)), public elemmod::grid\_maxy**

Maximaler Gitterabstand in y-Richtung.

**3.9.2.15 REAL(KIND(0D0)), public elemmod::grid\_min**

Minaler Gitterabstand (Betrag)

**3.9.2.16 REAL(KIND(0D0)), public elemmod::grid\_minx**

Minimaler Gitterabstand in x-Richtung.

**3.9.2.17 REAL(KIND(0D0)), public elemmod::grid\_miny**

Minimaler Gitterabstand in y-Richtung.

**3.9.2.18 LOGICAL, public elemmod::lrandb2**

switch boundary values



**3.9.2.19 LOGICAL, public elemmod::lsink**

switch/number fictitious sink node (only for 2D)

**3.9.2.20 INTEGER(KIND = 4), public elemmod::mb**

Bandbreite der Gesamtsteifigkeitsmatrix 'a'.

**3.9.2.21 INTEGER, dimension(:,:), allocatable, public elemmod::nachbar**

Zeiger auf die Nachbarn der nichtentarteten Elemente.

**3.9.2.22 INTEGER(KIND = 4), dimension(:), allocatable, public elemmod::nelanz**

Anzahl der Elemente eines bestimmten Typs.

**3.9.2.23 INTEGER(KIND = 4), dimension(:,:), allocatable, public elemmod::nrel**

Knotennummern der Elemente (Reihenfolge !)

**3.9.2.24 INTEGER(KIND = 4), public elemmod::nsink**

number of grid node for sink

**3.9.2.25 INTEGER(KIND = 4), public elemmod::relanz**

Anzahl der Randelemente.

**3.9.2.26 INTEGER(KIND = 4), dimension(:), allocatable, public elemmod::rnr**

Zeiger auf Werte der Randelemente.

**3.9.2.27 INTEGER(KIND = 4), public elemmod::sanz**

Anzahl der Knoten (bzw. Knotenvariablen)

**3.9.2.28 INTEGER(KIND = 4), dimension(:), allocatable, public elemmod::selanz**

Anzahl der Knoten (bzw. Knotenvariablen) in einem Elementtyp.

**3.9.2.29 INTEGER(KIND = 4), public elemmod::smaxs**

Groeste Anzahl der Knoten der Flaechelemente.

**3.9.2.30 INTEGER(KIND = 4), dimension(:), allocatable, public elemmod::snr**

Zeiger auf Koordinaten der Knoten (Inverser Permutationsvektor der Umnummerierung)

**3.9.2.31 REAL(KIND(0D0)), dimension(:), allocatable, public elemmod::sx**

x-Koordinaten der Knoten

**3.9.2.32 REAL(KIND(0D0)), dimension(:), allocatable, public elemmod::sy**

y-Koordinaten der Knoten

**3.9.2.33 REAL(KIND(0D0)), public elemmod::sytop**

mittlere y-Koordinate aller Randelemente vom Typ 12 ("no flow")

**3.9.2.34 INTEGER(KIND = 4), dimension(:), allocatable, public elemmod::typ**

Elementtypen (Randelemente (ntyp > 10) am Schluss !)

**3.9.2.35 INTEGER(KIND = 4), public elemmod::typanz**

Anzahl der Elementtypen.

**3.9.2.36 REAL(KIND(0D0)), dimension(:), allocatable, public elemmod::xk**

x-Koordinaten der Eckknotenpunkte

**3.9.2.37 REAL(KIND(0D0)), dimension(:), allocatable, public elemmod::yk**

y-Koordinaten der Eckknotenpunkte

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

- [elemmod.f90](#)

## 3.10 errmod Module Reference

### Public Attributes

- INTEGER(KIND=4) [errnr](#)
- INTEGER(KIND=4) [fperr](#)
- INTEGER(KIND=4) [fprun](#)
- INTEGER(KIND=4) [fpinv](#)
- INTEGER(KIND=4) [fpcjg](#)
- INTEGER(KIND=4) [fpeps](#)
- INTEGER(KIND=4) [fpcfg](#)
- INTEGER(KIND=4) [errflag](#)
- CHARACTER(256), public [fetxt](#)

### 3.10.1 Member Data Documentation

3.10.1.1 INTEGER(KIND = 4) [errmod::errflag](#)

3.10.1.2 INTEGER(KIND = 4) [errmod::errnr](#)

3.10.1.3 CHARACTER (256), public [errmod::fetxt](#)

3.10.1.4 INTEGER(KIND = 4) [errmod::fpcfg](#)

3.10.1.5 INTEGER(KIND = 4) [errmod::fpcjg](#)

3.10.1.6 INTEGER(KIND = 4) [errmod::fpeps](#)

3.10.1.7 INTEGER(KIND = 4) [errmod::fperr](#)

3.10.1.8 INTEGER(KIND = 4) [errmod::fpinv](#)

3.10.1.9 INTEGER(KIND = 4) [errmod::fprun](#)

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

- [errmod.f90](#)

## 3.11 femmod Module Reference

### Public Attributes

- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [pot](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [pota](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [b](#)

- REAL(KIND(0D0)), dimension(:), allocatable, public [bdc](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [fak](#)
- REAL(KIND(0D0)), dimension(:, :, :), allocatable, public [elbg](#)
- REAL(KIND(0D0)), dimension(:, :, :), allocatable, public [relbg](#)
- REAL(KIND(0D0)), dimension(:, :, :), allocatable, public [kg](#)
- LOGICAL, public [lbeta](#)
- LOGICAL, public [lrandb](#)
- LOGICAL, public [lsr](#)
- LOGICAL, public [ldc](#)

### 3.11.1 Member Data Documentation

3.11.1.1 COMPLEX (KIND(0D0)), dimension(:), allocatable, public femmod::b

3.11.1.2 REAL (KIND(0D0)), dimension(:), allocatable, public femmod::bdc

3.11.1.3 REAL (KIND(0D0)), dimension(:, :, :), allocatable, public femmod::elbg

3.11.1.4 REAL (KIND(0D0)), dimension(:), allocatable, public femmod::fak

3.11.1.5 REAL (KIND(0D0)), dimension(:, :, :), allocatable, public femmod::kg

3.11.1.6 LOGICAL, public femmod::lbeta

3.11.1.7 LOGICAL, public femmod::ldc

3.11.1.8 LOGICAL, public femmod::lrandb

3.11.1.9 LOGICAL, public femmod::lsr

3.11.1.10 COMPLEX (KIND(0D0)), dimension(:), allocatable, public femmod::pot

3.11.1.11 COMPLEX (KIND(0D0)), dimension(:), allocatable, public femmod::pota

3.11.1.12 REAL (KIND(0D0)), dimension(:, :, :), allocatable, public femmod::relbg

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

- [femmod.f90](#)

## 3.12 get\_ver Module Reference

### Public Member Functions

- subroutine, public [get\\_git\\_ver](#)

### Public Attributes

- CHARACTER(256), dimension(5), public [version](#)

### 3.12.1 Member Function/Subroutine Documentation

- 3.12.1.1 subroutine, public [get\\_ver::get\\_git\\_ver](#) ( )

### 3.12.2 Member Data Documentation

- 3.12.2.1 CHARACTER (256), dimension(5), public [get\\_ver::version](#)

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

- [get\\_git\\_ver.f90](#)

## 3.13 invhpmo Module Reference

### Public Attributes

- INTEGER(KIND=4) [kanal](#)
- CHARACTER(128) [delem](#)
- CHARACTER(128) [delectr](#)
- CHARACTER(128) [dstrom](#)
- CHARACTER(128) [dsigma](#)
- CHARACTER(128) [dvolt](#)
- CHARACTER(128) [dsens](#)
- CHARACTER(128) [dstart](#)
- LOGICAL [lsetup](#)
- LOGICAL [lsetip](#)
- LOGICAL [lagain](#)
- INTEGER(KIND=4) [j](#)
- INTEGER(KIND=4) [k](#)
- INTEGER(KIND=4) [l](#)
- REAL [izeit](#)
- REAL, dimension(2) [tazeit](#)
- INTEGER(KIND=4) [errnr2](#)

### 3.13.1 Member Data Documentation

- 3.13.1.1 CHARACTER (128) [invhpmo::delectr](#)

- 3.13.1.2 CHARACTER (128) [invhpmo::delem](#)

- 3.13.1.3 CHARACTER (128) `invhpmo::dsens`
- 3.13.1.4 CHARACTER (128) `invhpmo::dsigma`
- 3.13.1.5 CHARACTER (128) `invhpmo::dstart`
- 3.13.1.6 CHARACTER (128) `invhpmo::dstrom`
- 3.13.1.7 CHARACTER (128) `invhpmo::dvolt`
- 3.13.1.8 INTEGER (KIND=4) `invhpmo::errnr2`
- 3.13.1.9 REAL `invhpmo::izeit`
- 3.13.1.10 INTEGER (KIND=4) `invhpmo::j`
- 3.13.1.11 INTEGER (KIND=4) `invhpmo::k`
- 3.13.1.12 INTEGER (KIND=4) `invhpmo::kanal`
- 3.13.1.13 INTEGER (KIND=4) `invhpmo::l`
- 3.13.1.14 LOGICAL `invhpmo::lagain`
- 3.13.1.15 LOGICAL `invhpmo::lsetip`
- 3.13.1.16 LOGICAL `invhpmo::lsetup`
- 3.13.1.17 REAL, dimension(2) `invhpmo::tazeit`

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

- [invhpmo.f90](#)

## 3.14 invmod Module Reference

### Public Attributes

- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [dat](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [par](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [dpar](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [dpar2](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [d0](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [m0](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [fm0](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [m\\_ref](#)

- REAL(KIND(0D0)), dimension(:), allocatable, public [wmatd](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [wmatd2](#)
- REAL(KIND(0D0)), public [par\\_vari](#)
- INTEGER(KIND=4), dimension(:), allocatable, public [wdfak](#)
- LOGICAL(KIND=4), public [lfpi](#)
- INTEGER(KIND=4), dimension(:), allocatable, public [wmfak](#)

### 3.14.1 Member Data Documentation

3.14.1.1 COMPLEX(KIND(0D0)), dimension(:), allocatable, public `invmod::d0`

3.14.1.2 COMPLEX(KIND(0D0)), dimension(:), allocatable, public `invmod::dat`

3.14.1.3 COMPLEX(KIND(0D0)), dimension(:), allocatable, public `invmod::dpar`

3.14.1.4 COMPLEX(KIND(0D0)), dimension(:), allocatable, public `invmod::dpar2`

3.14.1.5 COMPLEX(KIND(0D0)), dimension(:), allocatable, public `invmod::fm0`

3.14.1.6 LOGICAL(KIND = 4), public `invmod::lfpi`

3.14.1.7 COMPLEX(KIND(0D0)), dimension(:), allocatable, public `invmod::m0`

3.14.1.8 COMPLEX(KIND(0D0)), dimension(:), allocatable, public `invmod::m_ref`

3.14.1.9 COMPLEX(KIND(0D0)), dimension(:), allocatable, public `invmod::par`

3.14.1.10 REAL(KIND(0D0)), public `invmod::par_vari`

3.14.1.11 INTEGER(KIND = 4), dimension(:), allocatable, public `invmod::wdfak`

3.14.1.12 REAL(KIND(0D0)), dimension(:), allocatable, public `invmod::wmatd`

3.14.1.13 REAL(KIND(0D0)), dimension(:), allocatable, public `invmod::wmatd2`

3.14.1.14 INTEGER (KIND = 4), dimension(:), allocatable, public `invmod::wmfak`

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

- [invmod.f90](#)

## 3.15 konvmod Module Reference

### Public Attributes

- REAL(KIND(0D0)), public [lam](#)

- REAL(KIND(0D0)), public [lammax](#)
- REAL(KIND(0D0)), public [lam\\_cri](#)
- REAL(KIND(0D0)), public [lamnull\\_cri](#)
- REAL(KIND(0D0)), public [lamnull\\_fpi](#)
- REAL(KIND(0D0)), public [lam\\_ref](#)
- INTEGER, public [lam\\_ref\\_sw](#)
- REAL(KIND(0D0)), public [lamfix](#)
- REAL(KIND(0D0)), public [dlam](#)
- REAL(KIND(0D0)), public [dlalt](#)
- INTEGER(KIND=4), public [nlam](#)
- REAL(KIND(0D0)), public [alam](#)
- REAL(KIND(0D0)), public [fstart](#)
- REAL(KIND(0D0)), public [fstop](#)
- REAL(KIND(0D0)), public [step](#)
- REAL(KIND(0D0)), public [stpalt](#)
- REAL(KIND(0D0)), public [stpmin](#)
- LOGICAL, public [llam](#)
- LOGICAL, public [lstep](#)
- LOGICAL, public [ldlami](#)
- LOGICAL, public [ldlamf](#)
- INTEGER, public [llamf](#)
- LOGICAL, public [lfstep](#)
- REAL(KIND(0D0)), public [nrmsd](#)
- REAL(KIND(0D0)), public [rmsalt](#)
- REAL(KIND(0D0)), public [rmsreg](#)
- REAL(KIND(0D0)), public [rmssum](#)
- REAL(KIND(0D0)), public [betrms](#)
- REAL(KIND(0D0)), public [pharms](#)
- REAL(KIND(0D0)), public [rough](#)
- REAL(KIND(0D0)), public [nrmsdm](#)
- REAL(KIND(0D0)), public [mqrms](#)
- REAL(KIND(0D0)), public [l1min](#)
- REAL(KIND(0D0)), public [l1rat](#)
- REAL(KIND(0D0)), public [bdpar](#)
- REAL(KIND(0D0)), public [badmin](#)
- INTEGER(KIND=4), public [nx](#)
- INTEGER(KIND=4), public [nz](#)
- INTEGER(KIND=4), public [itmax](#)
- INTEGER(KIND=4), public [it](#)
- INTEGER(KIND=4), public [itr](#)
- REAL(KIND(0D0)), public [alfx](#)
- REAL(KIND(0D0)), public [alfz](#)
- REAL(KIND(0D0)), public [betamgs](#)
- LOGICAL, public [lrobust](#)
- LOGICAL, public [ldiff](#)
- LOGICAL, public [lphi0](#)



- LOGICAL, public [lfphai](#)
- LOGICAL, public [lffhom](#)
- INTEGER(KIND=4), public [ltri](#)
- LOGICAL, public [lprior](#)
- LOGICAL, public [lw\\_ref](#)
- LOGICAL, public [lnsepri](#)
- LOGICAL, public [lsens](#)
- LOGICAL, public [lres](#)
- LOGICAL, public [lcov1](#)
- LOGICAL, public [lcov2](#)
- INTEGER(KIND=4), public [mswitch](#)
- LOGICAL, public [lgauss](#)
- LOGICAL, public [lvario](#)
- LOGICAL, public [lverb](#)
- LOGICAL, public [lverb\\_dat](#)
- LOGICAL, public [lsytop](#)
- LOGICAL, public [lelerr](#)

### 3.15.1 Member Data Documentation

3.15.1.1 REAL(KIND(0D0)), public `konvmod::alam`

3.15.1.2 REAL(KIND(0D0)), public `konvmod::alfx`

3.15.1.3 REAL(KIND(0D0)), public `konvmod::alfz`

3.15.1.4 REAL(KIND(0D0)), public `konvmod::badmin`

3.15.1.5 REAL(KIND(0D0)), public `konvmod::bdpar`

3.15.1.6 REAL(KIND(0D0)), public `konvmod::betamgs`

3.15.1.7 REAL(KIND(0D0)), public `konvmod::betrms`

3.15.1.8 REAL(KIND(0D0)), public `konvmod::dlalt`

3.15.1.9 REAL(KIND(0D0)), public `konvmod::dlam`

3.15.1.10 REAL(KIND(0D0)), public `konvmod::fstart`

3.15.1.11 REAL(KIND(0D0)), public `konvmod::fstop`

3.15.1.12 INTEGER(KIND = 4), public `konvmod::it`

3.15.1.13 INTEGER(KIND = 4), public `konvmod::itmax`

- 3.15.1.14 INTEGER(KIND = 4), public konvmod::itr
- 3.15.1.15 REAL(KIND(0D0)), public konvmod::l1min
- 3.15.1.16 REAL(KIND(0D0)), public konvmod::l1rat
- 3.15.1.17 REAL(KIND(0D0)), public konvmod::lam
- 3.15.1.18 REAL(KIND(0D0)), public konvmod::lam\_cri
- 3.15.1.19 REAL(KIND(0D0)), public konvmod::lam\_ref
- 3.15.1.20 INTEGER, public konvmod::lam\_ref\_sw
- 3.15.1.21 REAL(KIND(0D0)), public konvmod::lamfix
- 3.15.1.22 REAL(KIND(0D0)), public konvmod::lammax
- 3.15.1.23 REAL(KIND(0D0)), public konvmod::lamnull\_cri
- 3.15.1.24 REAL(KIND(0D0)), public konvmod::lamnull\_fpi
- 3.15.1.25 LOGICAL, public konvmod::lcov1
- 3.15.1.26 LOGICAL, public konvmod::lcov2
- 3.15.1.27 LOGICAL, public konvmod::ldiff
- 3.15.1.28 LOGICAL, public konvmod::ldlamf
- 3.15.1.29 LOGICAL, public konvmod::ldlami
- 3.15.1.30 LOGICAL, public konvmod::lelerr
- 3.15.1.31 LOGICAL, public konvmod::lffhom
- 3.15.1.32 LOGICAL, public konvmod::lfphai
- 3.15.1.33 LOGICAL, public konvmod::lfstep
- 3.15.1.34 LOGICAL, public konvmod::lgauss
- 3.15.1.35 LOGICAL, public konvmod::llam
- 3.15.1.36 INTEGER, public konvmod::llamf
- 3.15.1.37 LOGICAL, public konvmod::linsepri

- 3.15.1.38 LOGICAL, public konvmod::lphi0
- 3.15.1.39 LOGICAL, public konvmod::lprior
- 3.15.1.40 LOGICAL, public konvmod::lres
- 3.15.1.41 LOGICAL, public konvmod::lrobust
- 3.15.1.42 LOGICAL, public konvmod::lsens
- 3.15.1.43 LOGICAL, public konvmod::lstep
- 3.15.1.44 LOGICAL, public konvmod::lsytop
- 3.15.1.45 INTEGER(KIND = 4), public konvmod::ltri
- 3.15.1.46 LOGICAL, public konvmod::lvario
- 3.15.1.47 LOGICAL, public konvmod::lverb
- 3.15.1.48 LOGICAL, public konvmod::lverb\_dat
- 3.15.1.49 LOGICAL, public konvmod::lw\_ref
- 3.15.1.50 REAL(KIND(OD0)), public konvmod::mqrms
- 3.15.1.51 INTEGER(KIND = 4), public konvmod::mswitch
- 3.15.1.52 INTEGER(KIND = 4), public konvmod::nlam
- 3.15.1.53 REAL(KIND(OD0)), public konvmod::nrmsd
- 3.15.1.54 REAL(KIND(OD0)), public konvmod::nrmsdm
- 3.15.1.55 INTEGER(KIND = 4), public konvmod::nx
- 3.15.1.56 INTEGER(KIND = 4), public konvmod::nz
- 3.15.1.57 REAL(KIND(OD0)), public konvmod::pharms
- 3.15.1.58 REAL(KIND(OD0)), public konvmod::rmsalt
- 3.15.1.59 REAL(KIND(OD0)), public konvmod::rmsreg
- 3.15.1.60 REAL(KIND(OD0)), public konvmod::rmssum
- 3.15.1.61 REAL(KIND(OD0)), public konvmod::rough

3.15.1.62 REAL(KIND(0D0)), public konvmod::step

3.15.1.63 REAL(KIND(0D0)), public konvmod::stpalt

3.15.1.64 REAL(KIND(0D0)), public konvmod::stpmin

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

- [konvmod.f90](#)

## 3.16 Make\_noise Module Reference

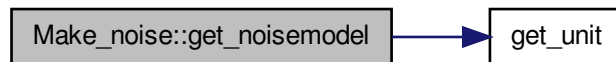
### Public Member Functions

- subroutine, public [get\\_noisemodel](#) (iseed, wa, w0, pa1, pb, pa2, p0, ierr)
- subroutine, public [write\\_noisemodel](#) (iseed, wa, w0, pa1, pb, pa2, p0, ierr)
- subroutine, public [Random\\_Init](#) (iseed)
- REAL(KIND(0D0)) function, public [Random\\_Gauss](#) (mean, sigma)
- REAL(KIND(0D0)) function, public [Random\\_Exponential](#) (tau, tmin, tmax)
- REAL(KIND(0D0)) function, public [Random\\_BreitWigner](#) (mean, fwhm)

### 3.16.1 Member Function/Subroutine Documentation

3.16.1.1 subroutine, public **Make\_noise::get\_noisemodel** ( INTEGER ( KIND = 4 ),  
intent(inout) *iseed*, REAL(KIND(0D0)), intent(inout) *wa*, REAL(KIND(0D0)), intent(inout)  
*w0*, REAL(KIND(0D0)), intent(inout) *pa1*, REAL(KIND(0D0)), intent(inout) *pb*,  
REAL(KIND(0D0)), intent(inout) *pa2*, REAL(KIND(0D0)), intent(inout) *p0*, INTEGER *ierr* )

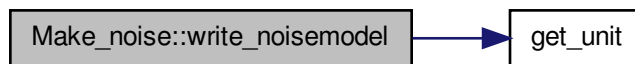
Here is the call graph for this function:



3.16.1.2 REAL ( KIND ( 0D0 )) function, public **Make\_noise::Random\_BreitWigner** ( REAL ( KIND  
(0D0)), intent(in), optional *mean*, REAL ( KIND ( 0D0 )), intent(in), optional *fwhm* )

- 3.16.1.3 REAL (KIND (0D0)) function, public Make\_noise::Random\_Exponential ( REAL (KIND (0D0)), intent(in) *tau*, REAL (KIND (0D0)), intent(in), optional *tmin*, REAL (KIND (0D0)), intent(in), optional *tmax* )
- 3.16.1.4 REAL (KIND (0D0)) function, public Make\_noise::Random\_Gauss ( INTEGER, intent(in), optional *mean*, INTEGER, intent(in), optional *sigma* )
- 3.16.1.5 subroutine, public Make\_noise::Random\_Init ( INTEGER ( KIND = 4 ), optional *iseed* )
- 3.16.1.6 subroutine, public Make\_noise::write\_noisemodel ( INTEGER ( KIND = 4 ), intent(in) *iseed*, REAL(KIND(0D0)), intent(in) *wa*, REAL(KIND(0D0)), intent(in) *w0*, REAL(KIND(0D0)), intent(in) *pa1*, REAL(KIND(0D0)), intent(in) *pb*, REAL(KIND(0D0)), intent(in) *pa2*, REAL(KIND(0D0)), intent(in) *p0*, INTEGER *ierr* )

Here is the call graph for this function:



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

- [make\\_noise.f90](#)

## 3.17 modelmod Module Reference

### Public Attributes

- INTEGER(KIND=4), public [manz](#)
- INTEGER(KIND=4), dimension(:), allocatable, public [mnr](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [w\\_ref\\_re](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [w\\_ref\\_im](#)
- INTEGER, dimension(:), allocatable, public [ind\\_ref\\_grad](#)

### 3.17.1 Member Data Documentation

- 3.17.1.1 INTEGER, dimension(:), allocatable, public modelmod::ind\_ref\_grad

- 3.17.1.2 `INTEGER(KIND = 4), public modelmod::manz`
- 3.17.1.3 `INTEGER(KIND = 4), dimension(:), allocatable, public modelmod::mnr`
- 3.17.1.4 `REAL(KIND(0D0)), dimension(:), allocatable, public modelmod::w_ref_im`
- 3.17.1.5 `REAL(KIND(0D0)), dimension(:), allocatable, public modelmod::w_ref_re`

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

- [modelmod.f90](#)

## 3.18 ompmod Module Reference

### Public Attributes

- `INTEGER, public TID`
- `INTEGER, public NTHREADS`
- `INTEGER, parameter, public CHUNK_0 = 256`
- `INTEGER, parameter, public CHUNK_1 = 2*CHUNK_0`
- `INTEGER, parameter, public CHUNK_2 = 2*CHUNK_1`
- `INTEGER, parameter, public CHUNK_3 = 2*CHUNK_2`

### 3.18.1 Member Data Documentation

- 3.18.1.1 `INTEGER, parameter, public ompmod::CHUNK_0 = 256`
- 3.18.1.2 `INTEGER, parameter, public ompmod::CHUNK_1 = 2*CHUNK_0`
- 3.18.1.3 `INTEGER, parameter, public ompmod::CHUNK_2 = 2*CHUNK_1`
- 3.18.1.4 `INTEGER, parameter, public ompmod::CHUNK_3 = 2*CHUNK_2`
- 3.18.1.5 `INTEGER, public ompmod::NTHREADS`
- 3.18.1.6 `INTEGER, public ompmod::TID`

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

- [ompmod.f90](#)

## 3.19 pathmod Module Reference

### Public Member Functions

- subroutine, public [clear\\_string](#) (string)

### Public Attributes

- CHARACTER(1), public [slash](#)
- CHARACTER(60), public [ramd](#)
- INTEGER(KIND=4), public [lnramd](#)
- CHARACTER(6), parameter, public [mkdir](#) = 'mkdir '
- CHARACTER(6), parameter, public [rmdir](#) = 'rm -R '

#### 3.19.1 Member Function/Subroutine Documentation

3.19.1.1 subroutine, public [pathmod::clear\\_string](#) ( CHARACTER(\*) *string* )

#### 3.19.2 Member Data Documentation

3.19.2.1 INTEGER(KIND = 4), public [pathmod::lnramd](#)

3.19.2.2 CHARACTER(6), parameter, public [pathmod::mkdir](#) = 'mkdir '

3.19.2.3 CHARACTER (60), public [pathmod::ramd](#)

3.19.2.4 CHARACTER(6), parameter, public [pathmod::rmdir](#) = 'rm -R '

3.19.2.5 CHARACTER(1), public [pathmod::slash](#)

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

- [pathmod.f90](#)

## 3.20 randbmod Module Reference

### Public Attributes

- INTEGER(KIND=4), public [rwdanz](#)
- INTEGER(KIND=4), dimension(:), allocatable, public [rwdnr](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [rwddc](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [rwd](#)
- INTEGER(KIND=4), public [rwdbnr](#)
- INTEGER(KIND=4), public [rwnanz](#)
- REAL(KIND(0D0)), dimension(:), allocatable, public [rwndc](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [rwn](#)

### 3.20.1 Member Data Documentation

3.20.1.1 COMPLEX(KIND(0D0)), dimension(:), allocatable, public randbmod::rwd

3.20.1.2 INTEGER(KIND = 4), public randbmod::rwdanz

3.20.1.3 INTEGER(KIND = 4), public randbmod::rwdbnr

3.20.1.4 REAL(KIND(0D0)), dimension(:), allocatable, public randbmod::rwdde

3.20.1.5 INTEGER(KIND = 4), dimension(:), allocatable, public randbmod::rwdnr

3.20.1.6 COMPLEX(KIND(0D0)), dimension(:), allocatable, public randbmod::rwn

3.20.1.7 INTEGER(KIND = 4), public randbmod::rwnanz

3.20.1.8 REAL(KIND(0D0)), dimension(:), allocatable, public randbmod::rwnde

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

- [randbmod.f90](#)

## 3.21 sigmamod Module Reference

### Public Attributes

- COMPLEX(KIND(0D0)), public [sigma0](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [sigma](#)
- COMPLEX(KIND(0D0)), dimension(:), allocatable, public [sigma2](#)
- REAL(KIND(0D0)), public [bet0](#)
- REAL(KIND(0D0)), public [pha0](#)
- LOGICAL, public [lrho0](#)
- LOGICAL, public [lstart](#)
- INTEGER(KIND=4), public [iseedpri](#)
- REAL(KIND(0D0)), public [modl\\_stdn](#)

### 3.21.1 Member Data Documentation

3.21.1.1 REAL(KIND(0D0)), public sigmamod::bet0

3.21.1.2 INTEGER(KIND = 4), public sigmamod::iseedpri

3.21.1.3 LOGICAL, public sigmamod::lrho0

3.21.1.4 LOGICAL, public sigmamod::lstart



3.21.1.5 REAL(KIND(0D0)), public sigmamod::modl\_stdn

3.21.1.6 REAL(KIND(0D0)), public sigmamod::pha0

3.21.1.7 COMPLEX(KIND(0D0)), dimension(:), allocatable, public sigmamod::sigma

3.21.1.8 COMPLEX(KIND(0D0)), public sigmamod::sigma0

3.21.1.9 COMPLEX(KIND(0D0)), dimension(:), allocatable, public sigmamod::sigma2

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

- [sigmamod.f90](#)

## 3.22 tic\_toc Module Reference

### Public Member Functions

- subroutine, public [tic](#) (c1)
- subroutine, public [toc](#) (c1, csz)

### 3.22.1 Member Function/Subroutine Documentation

3.22.1.1 subroutine, public tic\_toc::tic ( INTEGER(KIND = 4), intent(out) c1 )

3.22.1.2 subroutine, public tic\_toc::toc ( INTEGER(KIND = 4), intent(in) c1, CHARACTER (\*), intent(inout) csz )

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

- [tic\\_toc.f90](#)

## 3.23 variomodel Module Reference

### Public Member Functions

- subroutine, public [set\\_vario](#) (type, ax, ay, esp\_mit, esp\_med)
- subroutine, public [get\\_vario](#) (ax, ay, csz, type)
- REAL(KIND(0D0)) function, public [mvario](#) (lagx, lagy, varianz)
- REAL(KIND(0D0)) function, public [mcova](#) (lagx, lagy, varianz)

### 3.23.1 Member Function/Subroutine Documentation

- 3.23.1.1 subroutine, public `variomodel::get_vario` ( `REAL (KIND(0D0))`, intent(out) *ax*, `REAL (KIND(0D0))`, intent(out) *ay*, `CHARACTER (*)` *csz*, `INTEGER`, intent(in) *type* )
- 3.23.1.2 `REAL (KIND (0D0))` function, public `variomodel::mcova` ( `REAL (KIND (0D0))`, intent(in) *lagx*, `REAL (KIND (0D0))`, intent(in) *lagy*, `REAL (KIND (0D0))`, intent(in) *varianz* )
- 3.23.1.3 `REAL (KIND (0D0))` function, public `variomodel::mvario` ( `REAL (KIND (0D0))`, intent(in) *lagx*, `REAL (KIND (0D0))`, intent(in) *lagy*, `REAL (KIND (0D0))`, intent(in) *varianz* )
- 3.23.1.4 subroutine, public `variomodel::set_vario` ( `INTEGER`, intent(in) *type*, `REAL(KIND(0D0))`, intent(in) *ax*, `REAL(KIND(0D0))`, intent(in) *ay*, `REAL(KIND(0D0))`, intent(in) *esp\_mit*, `REAL(KIND(0D0))`, intent(in) *esp\_med* )

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

- [variomodel.f90](#)

## 3.24 wavenmod Module Reference

### Public Attributes

- `INTEGER(KIND=4)`, public [kwnanz](#)
- `INTEGER(KIND=4)`, public [swrtr](#)
- `REAL(KIND(0D0))`, dimension(:), allocatable, public [kwn](#)
- `REAL(KIND(0D0))`, dimension(:), allocatable, public [kwnwi](#)
- `REAL(KIND(0D0))`, public [amin](#)
- `REAL(KIND(0D0))`, public [amax](#)

### 3.24.1 Member Data Documentation

- 3.24.1.1 `REAL(KIND(0D0))`, public `wavenmod::amax`
- 3.24.1.2 `REAL(KIND(0D0))`, public `wavenmod::amin`
- 3.24.1.3 `REAL(KIND(0D0))`, dimension(:), allocatable, public `wavenmod::kwn`
- 3.24.1.4 `INTEGER(KIND = 4)`, public `wavenmod::kwnanz`
- 3.24.1.5 `REAL(KIND(0D0))`, dimension(:), allocatable, public `wavenmod::kwnwi`
- 3.24.1.6 `INTEGER(KIND = 4)`, public `wavenmod::swrtr`

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

- [wavenmod.f90](#)



## Chapter 4

# File Documentation

### 4.1 alloci.f90 File Reference

#### Data Types

- module [alloci](#)

### 4.2 bbsedc.f90 File Reference

#### Functions/Subroutines

- subroutine [bbsedc](#) (kanal, datei)

#### 4.2.1 Function/Subroutine Documentation

4.2.1.1 subroutine [bbsedc](#) ( INTEGER (KIND = 4) *kanal*, CHARACTER (80) *datei* )

### 4.3 bbsens.f90 File Reference

#### Functions/Subroutines

- subroutine [bbsens](#) (kanal, datei)

#### 4.3.1 Function/Subroutine Documentation

4.3.1.1 subroutine [bbsens](#) ( INTEGER (KIND = 4) *kanal*, CHARACTER (80) *datei* )

## 4.4 bsp\_elem.f90 File Reference

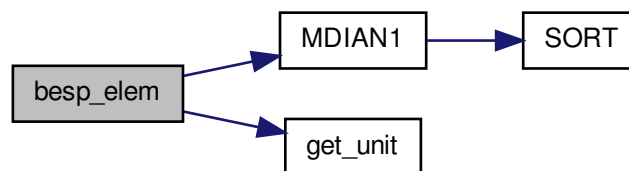
### Functions/Subroutines

- subroutine [bsp\\_elem](#)

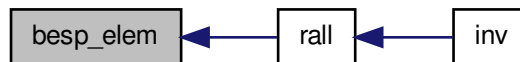
#### 4.4.1 Function/Subroutine Documentation

##### 4.4.1.1 subroutine [bsp\\_elem](#) ( )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.5 bessio.f90 File Reference

### Functions/Subroutines

- REAL(KIND(OD0)) function [BESSIO](#) (X)

#### 4.5.1 Function/Subroutine Documentation

4.5.1.1 REAL (KIND(0D0)) function BESSI0 ( REAL (KIND (0D0)) X )

### 4.6 bess1.f90 File Reference

#### Functions/Subroutines

- REAL(KIND(0D0)) function [BESSI1](#) (X)

#### 4.6.1 Function/Subroutine Documentation

4.6.1.1 REAL (KIND(0D0)) function BESSI1 ( REAL (KIND(0D0)) X )

### 4.7 bessk0.f90 File Reference

#### Functions/Subroutines

- REAL(KIND(0D0)) function [BESSK0](#) (X)

#### 4.7.1 Function/Subroutine Documentation

4.7.1.1 REAL (KIND(0D0)) function BESSK0 ( REAL (KIND(0D0)) X )

### 4.8 bessk1.f90 File Reference

#### Functions/Subroutines

- REAL(KIND(0D0)) function [BESSK1](#) (X)

#### 4.8.1 Function/Subroutine Documentation

4.8.1.1 REAL (KIND(0D0)) function BESSK1 ( REAL (KIND(0D0)) X )

### 4.9 beta.f90 File Reference

#### Functions/Subroutines

- REAL(KIND(0D0)) function [beta](#) (nelec, k)

### 4.9.1 Function/Subroutine Documentation

4.9.1.1 REAL (KIND(OD0)) function beta ( INTEGER (KIND = 4) *nelec*, INTEGER (KIND = 4) *k* )

## 4.10 bkfak.f90 File Reference

### Functions/Subroutines

- subroutine [bkfak](#) ()

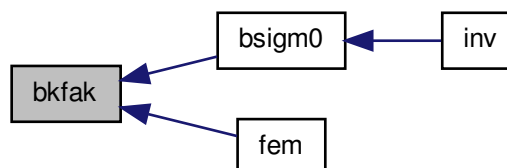
### 4.10.1 Function/Subroutine Documentation

4.10.1.1 subroutine [bkfak](#) ( )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.11 blam0.f90 File Reference



## Functions/Subroutines

- subroutine [blam0](#) ()

### 4.11.1 Function/Subroutine Documentation

#### 4.11.1.1 subroutine [blam0](#) ( )

Here is the caller graph for this function:



## 4.12 bmcm\_mod.f90 File Reference

### Data Types

- module [bmcm\\_mod](#)

## 4.13 bnachbar.f90 File Reference

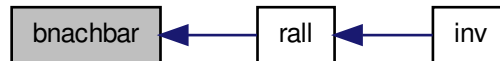
### Functions/Subroutines

- subroutine [bnachbar](#)

### 4.13.1 Function/Subroutine Documentation

#### 4.13.1.1 subroutine `bnachbar` ( )

Here is the caller graph for this function:



### 4.14 `bpar.f90` File Reference

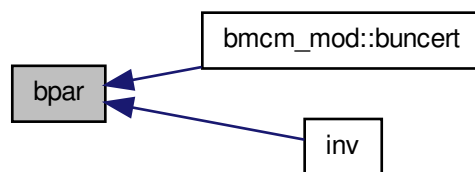
#### Functions/Subroutines

- subroutine `bpar`

#### 4.14.1 Function/Subroutine Documentation

##### 4.14.1.1 subroutine `bpar` ( )

Here is the caller graph for this function:



### 4.15 `bpot.f90` File Reference

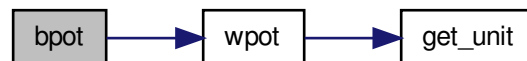
#### Functions/Subroutines

- subroutine `bpot` (kanal, datei)

### 4.15.1 Function/Subroutine Documentation

4.15.1.1 subroutine `bpot` ( `INTEGER (KIND = 4)` *kanal*, `CHARACTER (80)` *datei* )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.16 brough\_mod.f90 File Reference

### Data Types

- module [brough\\_mod](#)

## 4.17 bsendc.f90 File Reference

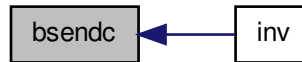
### Functions/Subroutines

- subroutine [bsendc](#) (tictoc)

### 4.17.1 Function/Subroutine Documentation

#### 4.17.1.1 subroutine `bsendc` ( LOGICAL *tictoc* )

Here is the caller graph for this function:



### 4.18 bsens.f90 File Reference

#### Functions/Subroutines

- subroutine `bsens` ( )

#### 4.18.1 Function/Subroutine Documentation

##### 4.18.1.1 subroutine `bsens` ( )

Here is the caller graph for this function:



### 4.19 bsensi.f90 File Reference

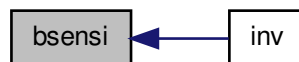
#### Functions/Subroutines

- subroutine `bsensi` (*tictoc*)

### 4.19.1 Function/Subroutine Documentation

#### 4.19.1.1 subroutine bsensi ( LOGICAL *tictoc* )

Here is the caller graph for this function:



## 4.20 bsign0.f90 File Reference

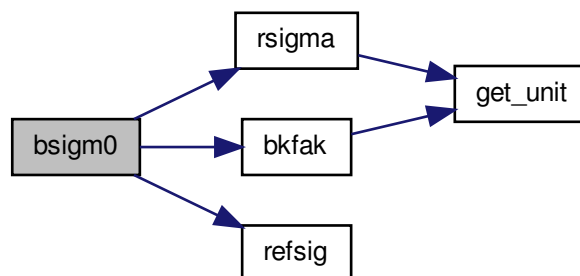
### Functions/Subroutines

- subroutine `bsigm0` (*kanal*, *dstart*)

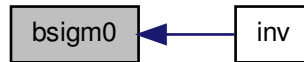
### 4.20.1 Function/Subroutine Documentation

#### 4.20.1.1 subroutine bsign0 ( INTEGER (KIND = 4) *kanal*, CHARACTER (80) *dstart* )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.21 bsigma.f90 File Reference

### Functions/Subroutines

- subroutine [bsigma](#)

### 4.21.1 Function/Subroutine Documentation

#### 4.21.1.1 subroutine [bsigma](#) ( )

Here is the caller graph for this function:



## 4.22 bsmatm\_mod.f90 File Reference

### Data Types

- module [bsmatm\\_mod](#)

## 4.23 bsytop.f90 File Reference

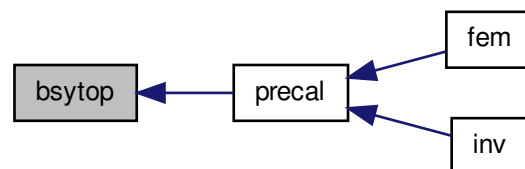
## Functions/Subroutines

- subroutine [bsytop](#)

### 4.23.1 Function/Subroutine Documentation

#### 4.23.1.1 subroutine [bsytop](#) ( )

Here is the caller graph for this function:



## 4.24 bvariogram.f90 File Reference

## Functions/Subroutines

- subroutine [bvariogram](#)

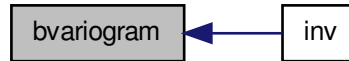
### 4.24.1 Function/Subroutine Documentation

#### 4.24.1.1 subroutine [bvariogram](#) ( )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.25 bvariogram.s.f90 File Reference

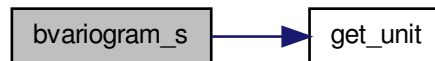
### Functions/Subroutines

- subroutine [bvariogram\\_s](#)

### 4.25.1 Function/Subroutine Documentation

#### 4.25.1.1 subroutine [bvariogram\\_s](#) ( )

Here is the call graph for this function:



## 4.26 bvolt.f90 File Reference

### Functions/Subroutines

- subroutine [bvolt](#) ( )

### 4.26.1 Function/Subroutine Documentation



## 4.26.1.1 subroutine bvolt ( )

Here is the caller graph for this function:



## 4.27 bvolti.f90 File Reference

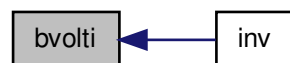
## Functions/Subroutines

- subroutine [bvolti](#) ( )

## 4.27.1 Function/Subroutine Documentation

## 4.27.1.1 subroutine bvolti ( )

Here is the caller graph for this function:



## 4.28 cg\_mod.f90 File Reference

## Data Types

- module [cg\\_mod](#)

## 4.29 chareal.f90 File Reference

### Functions/Subroutines

- REAL(KIND(OD0)) function `chareal` (txt, ltxt)

### 4.29.1 Function/Subroutine Documentation

4.29.1.1 REAL (KIND(OD0)) function `chareal` ( CHARACTER (\*) *txt*, INTEGER (KIND = 4) *ltxt* )

## 4.30 chkpol.f90 File Reference

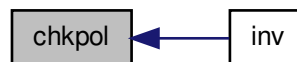
### Functions/Subroutines

- subroutine `chkpol` (lsetup)

### 4.30.1 Function/Subroutine Documentation

4.30.1.1 subroutine `chkpol` ( LOGICAL *lsetup* )

Here is the caller graph for this function:



## 4.31 chol.f90 File Reference

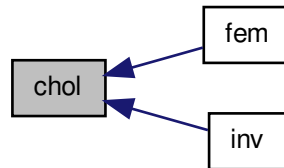
### Functions/Subroutines

- subroutine `chol` (a\_chol)

### 4.31.1 Function/Subroutine Documentation

4.31.1.1 subroutine `chol` ( `COMPLEX (KIND(OD0))`, `dimension(*) a_chol` )

Here is the caller graph for this function:



## 4.32 chold.f90 File Reference

### Functions/Subroutines

- subroutine `chold` (`a`, `p`, `n`, `ierr`, `lverb`)

#### 4.32.1 Function/Subroutine Documentation

4.32.1.1 subroutine `chold` ( `REAL (KIND(OD0))`, `dimension (n,n)`, `intent(inout) a`, `REAL (KIND(OD0))`, `dimension (n)`, `intent(out) p`, `INTEGER, intent(in) n`, `INTEGER, intent(out) ierr`, `LOGICAL, intent(in) lverb` )

## 4.33 choldc.f90 File Reference

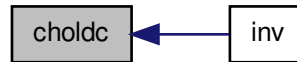
### Functions/Subroutines

- subroutine `choldc` (`a_chol`)

#### 4.33.1 Function/Subroutine Documentation

#### 4.33.1.1 subroutine `choldc` ( `REAL (KIND(OD0))`, `dimension(*) a_chol` )

Here is the caller graph for this function:



### 4.34 cholz.f90 File Reference

#### Functions/Subroutines

- subroutine `cholz` (`a`, `p`, `n`, `ierr`, `lverb`)

#### 4.34.1 Function/Subroutine Documentation

4.34.1.1 subroutine `cholz` ( `COMPLEX (KIND(OD0))`, `dimension (n,n)`, `intent(inout) a`, `COMPLEX (KIND(OD0))`, `dimension (n)`, `intent(out) p`, `INTEGER`, `intent(in) n`, `INTEGER`, `intent(out) ierr`, `LOGICAL`, `intent(in) lverb` )

### 4.35 cjgmod.f90 File Reference

#### Data Types

- module `cjgmod`

### 4.36 crerror.h File Reference

### 4.37 datmod.f90 File Reference

#### Data Types

- module `datmod`

## 4.38 dmisft.f90 File Reference

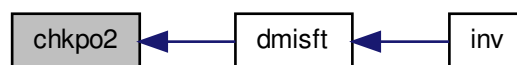
### Functions/Subroutines

- subroutine [dmisft](#) (*lsetup*)
- subroutine [chkpo2](#) (*dati*, *sigi*, *cdat*, *csig*, *wdlok*, *ldum*)

### 4.38.1 Function/Subroutine Documentation

4.38.1.1 subroutine **chkpo2** ( **COMPLEX (KIND(0D0))** *dati*, **COMPLEX (KIND(0D0))** *sigi*, **COMPLEX (KIND(0D0))** *cdat*, **COMPLEX (KIND(0D0))** *csig*, **INTEGER (KIND = 4)** *wdlok*, **LOGICAL** *ldum* )

Here is the caller graph for this function:



4.38.1.2 subroutine **dmisft** ( **LOGICAL** *lsetup* )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.39 electrmod.f90 File Reference

### Data Types

- module [electrmod](#)

## 4.40 elem1.f90 File Reference

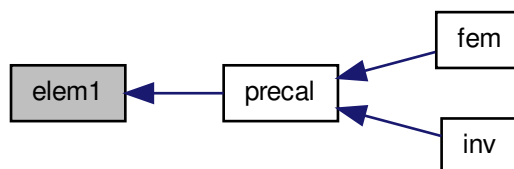
### Functions/Subroutines

- subroutine [elem1](#) ( )

### 4.40.1 Function/Subroutine Documentation

#### 4.40.1.1 subroutine `elem1` ( )

Here is the caller graph for this function:



## 4.41 elem3.f90 File Reference

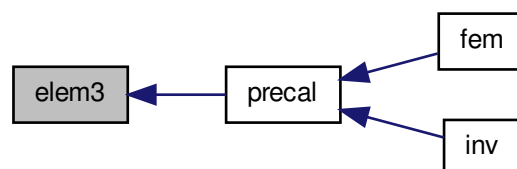
### Functions/Subroutines

- subroutine [elem3](#) ()

#### 4.41.1 Function/Subroutine Documentation

##### 4.41.1.1 subroutine [elem3](#) ( )

Here is the caller graph for this function:



## 4.42 elem4.f90 File Reference

### Functions/Subroutines

- subroutine [elem4](#) ()

#### 4.42.1 Function/Subroutine Documentation

##### 4.42.1.1 subroutine [elem4](#) ( )

## 4.43 elem5.f90 File Reference

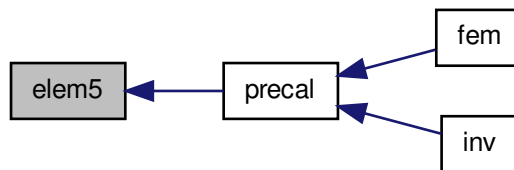
### Functions/Subroutines

- subroutine [elem5](#) ()

### 4.43.1 Function/Subroutine Documentation

#### 4.43.1.1 subroutine elem5 ( )

Here is the caller graph for this function:



## 4.44 elem8.f90 File Reference

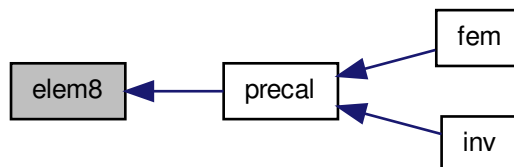
### Functions/Subroutines

- subroutine [elem8](#) (kelmas, kelve, kwert, smaxs)

### 4.44.1 Function/Subroutine Documentation

4.44.1.1 subroutine `elem8` ( `REAL(KIND(0D0))`, `dimension(smaxs,smaxs)` *kelmas*, `REAL(KIND(0D0))`, `dimension(smaxs)` *kelve*, `REAL(KIND(0D0))` *kwert*, `INTEGER (KIND = 4)` *smaxs* )

Here is the caller graph for this function:





## 4.45 elemmod.f90 File Reference

### Data Types

- module [elemmod](#)

*Replacement of former 'elem.fin' and basically contains the FE-element related variables and two methods for allocation and deallocation of global memory.*

## 4.46 errmod.f90 File Reference

### Data Types

- module [errmod](#)

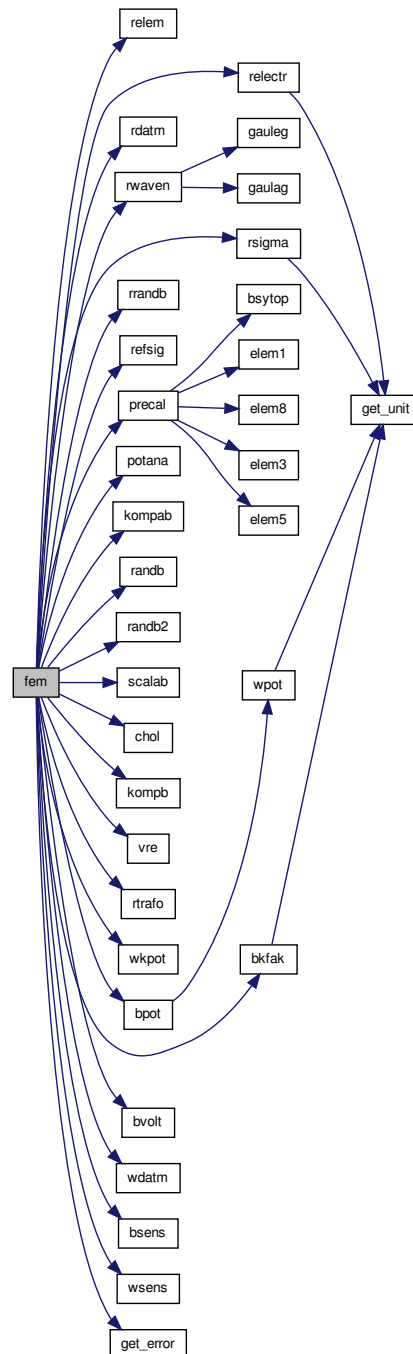
## 4.47 fem.f90 File Reference

### Functions/Subroutines

- program [fem](#)

### 4.47.1 Function/Subroutine Documentation

Here is the call graph for this function:



## 4.48 femmod.f90 File Reference

### Data Types

- module [femmod](#)

## 4.49 filpat.f90 File Reference

### Functions/Subroutines

- CHARACTER(\*) function [filpat](#) (disfile, ln, sw, slash)

### 4.49.1 Function/Subroutine Documentation

- 4.49.1.1 CHARACTER (\*) function [filpat](#) ( CHARACTER (\*) *disfile*, INTEGER (KIND = 4) *ln*, INTEGER (KIND = 4) *sw*, CHARACTER (1) *slash* )

## 4.50 gammln.f90 File Reference

### Functions/Subroutines

- REAL(KIND(0D0)) function [gammln](#) (xx)

### 4.50.1 Function/Subroutine Documentation

- 4.50.1.1 REAL (KIND(0D0)) function [gammln](#) ( REAL (KIND(0D0)) *xx* )

## 4.51 gaulag.f90 File Reference

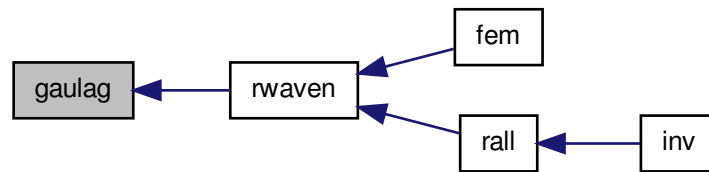
### Functions/Subroutines

- subroutine [gaulag](#) (x, w, n, alf)

### 4.51.1 Function/Subroutine Documentation

4.51.1.1 subroutine **gaulag** ( REAL (KIND(0D0)), dimension(n) *x*, REAL (KIND(0D0)), dimension(n) *w*, INTEGER (KIND = 4) *n*, REAL (KIND(0D0)) *alf* )

Here is the caller graph for this function:



## 4.52 gauleg.f90 File Reference

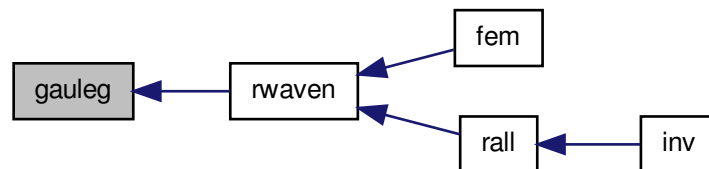
### Functions/Subroutines

- subroutine **gauleg** (*x1*, *x2*, *x*, *w*, *n*)

### 4.52.1 Function/Subroutine Documentation

4.52.1.1 subroutine **gauleg** ( REAL (KIND(0D0)) *x1*, REAL (KIND(0D0)) *x2*, REAL (KIND(0D0)), dimension(n) *x*, REAL (KIND(0D0)), dimension(n) *w*, INTEGER (KIND = 4) *n* )

Here is the caller graph for this function:



## 4.53 gauss\_cmplx.f90 File Reference

### Functions/Subroutines

- subroutine [Gauss\\_cmplx](#) (a, n, e\_flag)

#### 4.53.1 Function/Subroutine Documentation

- 4.53.1.1 subroutine [Gauss\\_cmplx](#) ( COMPLEX(KIND(0D0)), dimension(n,n), intent(inout) *a*,  
INTEGER(KIND(4)) *n*, INTEGER(KIND(4)) *e\_flag* )

## 4.54 gauss\_dble.f90 File Reference

### Functions/Subroutines

- subroutine [Gauss\\_dble](#) (a, n, e\_flag)

#### 4.54.1 Function/Subroutine Documentation

- 4.54.1.1 subroutine [Gauss\\_dble](#) ( REAL(KIND(0D0)), dimension(n,n), intent(inout) *a*,  
INTEGER(KIND(4)) *n*, INTEGER(KIND(4)) *e\_flag* )

## 4.55 get\_error.f90 File Reference

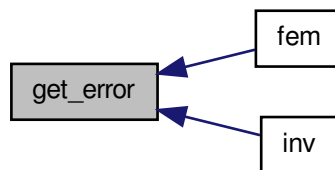
### Functions/Subroutines

- subroutine [get\\_error](#) (ftext, errnr, errflag, intext)

#### 4.55.1 Function/Subroutine Documentation

4.55.1.1 subroutine `get_error` ( CHARACTER(256), intent(out) *ftext*, INTEGER, intent(in) *errnr*,  
INTEGER, intent(in) *errflag*, CHARACTER(80), intent(in) *intext* )

Here is the caller graph for this function:



## 4.56 `get_git_ver.f90` File Reference

### Data Types

- module [get\\_ver](#)

## 4.57 `get_unit.f90` File Reference

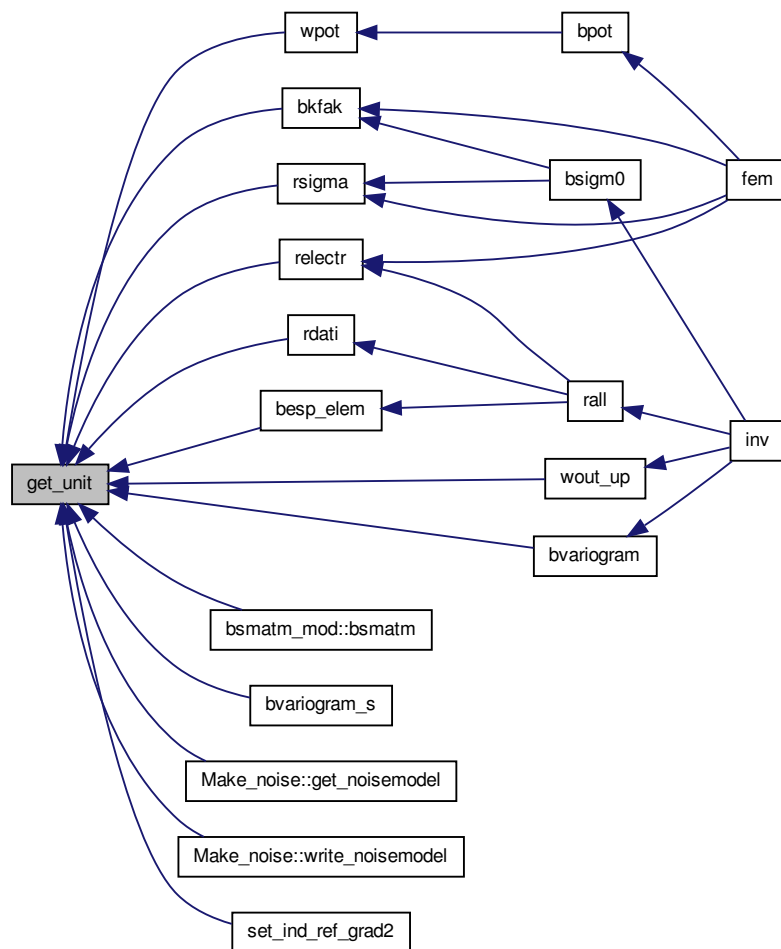
### Functions/Subroutines

- subroutine [get\\_unit](#) (iunit)
- subroutine [read\\_comments](#) (unit)

### 4.57.1 Function/Subroutine Documentation

## 4.57.1.1 subroutine get\_unit ( INTEGER ( kind = 4 ) iunit )

Here is the caller graph for this function:



#### 4.57.1.2 subroutine `read_comments` ( `INTEGER`, intent(in) *unit* )

Here is the caller graph for this function:



#### 4.58 `intcha.f90` File Reference

#### 4.59 `inv.f90` File Reference

#### Functions/Subroutines

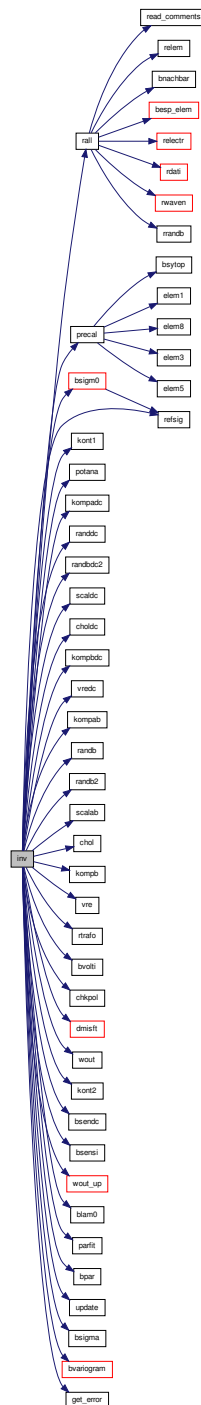
- program [inv](#)

#### 4.59.1 Function/Subroutine Documentation



## 4.59.1.1 program inv ( )

Here is the call graph for this function:



## 4.60 invhpmmod.f90 File Reference

### Data Types

- module [invhpmmod](#)

## 4.61 invmod.f90 File Reference

### Data Types

- module [invmod](#)

## 4.62 kompab.f90 File Reference

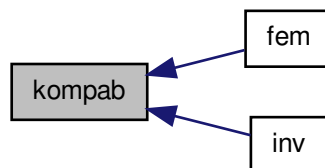
### Functions/Subroutines

- subroutine [kompab](#) (nelec, ki, my\_a, my\_b)

### 4.62.1 Function/Subroutine Documentation

4.62.1.1 subroutine **kompab** ( INTEGER (KIND = 4) *nelec*, INTEGER (KIND = 4) *ki*, COMPLEX (KIND (0D0)), dimension((mb+1)\*sanz) *my\_a*, COMPLEX (KIND (0D0)), dimension(sanz) *my\_b* )

Here is the caller graph for this function:



## 4.63 kompadc.f90 File Reference

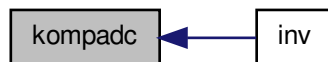
## Functions/Subroutines

- subroutine [kompadc](#) (*nelec*, *ki*, *a\_komp*, *b\_komp*)

### 4.63.1 Function/Subroutine Documentation

4.63.1.1 subroutine [kompadc](#) ( *INTEGER* (KIND = 4) *nelec*, *INTEGER* (KIND = 4) *ki*, *REAL* (KIND (0D0)), dimension((*mb* + 1)\**sanz*) *a\_komp*, *REAL* (KIND (0D0)), dimension(*sanz*) *b\_komp* )

Here is the caller graph for this function:



## 4.64 kompb.f90 File Reference

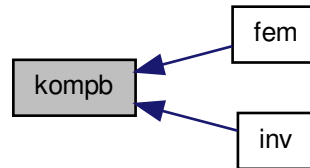
## Functions/Subroutines

- subroutine [kompb](#) (*nelec*, *b\_komp*, *fak\_komp*)

### 4.64.1 Function/Subroutine Documentation

4.64.1.1 subroutine **kompb** ( INTEGER (KIND = 4) *nelec*, COMPLEX (KIND(0D0)), dimension(\*)  
*b\_komp*, REAL (KIND(0D0)), dimension(\*) *fak\_komp* )

Here is the caller graph for this function:



## 4.65 kompbd.c.f90 File Reference

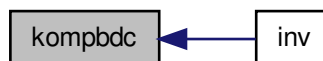
### Functions/Subroutines

- subroutine **kompbd** (*nelec*, *b\_komp*, *fak\_komp*)

### 4.65.1 Function/Subroutine Documentation

4.65.1.1 subroutine **kompbd** ( INTEGER (KIND = 4) *nelec*, REAL (KIND(0D0)), dimension  
 (*sanz*) *b\_komp*, REAL (KIND(0D0)), dimension (*sanz*) *fak\_komp* )

Here is the caller graph for this function:



## 4.66 kont1.f90 File Reference

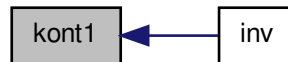
## Functions/Subroutines

- subroutine [kont1](#) (delem, delectr, dstrom, drandb, dd0, dm0, dfm0, lagain)

### 4.66.1 Function/Subroutine Documentation

4.66.1.1 subroutine `kont1` ( CHARACTER (80) *delem*, CHARACTER (80) *delectr*, CHARACTER (80) *dstrom*, CHARACTER (80) *drandb*, CHARACTER (80) *dd0*, CHARACTER (80) *dm0*, CHARACTER (80) *dfm0*, LOGICAL *lagain* )

Here is the caller graph for this function:



## 4.67 kont2.f90 File Reference

## Functions/Subroutines

- subroutine [kont2](#) (lsetup)

### 4.67.1 Function/Subroutine Documentation

4.67.1.1 subroutine `kont2` ( LOGICAL *lsetup* )

Here is the caller graph for this function:



## 4.68 konvmod.f90 File Reference

### Data Types

- module [konvmod](#)

## 4.69 linvd.f90 File Reference

### Functions/Subroutines

- subroutine [linvd](#) (a, p, n, lverb)

### 4.69.1 Function/Subroutine Documentation

4.69.1.1 subroutine [linvd](#) ( REAL (KIND(0D0)), dimension (n,n), intent(inout) *a*, REAL (KIND(0D0)), dimension (n), intent(in) *p*, INTEGER, intent(in) *n*, LOGICAL, intent(in) *lverb* )

## 4.70 linvz.f90 File Reference

### Functions/Subroutines

- subroutine [linvz](#) (a, p, n)

### 4.70.1 Function/Subroutine Documentation

4.70.1.1 subroutine [linvz](#) ( COMPLEX (KIND(0D0)), dimension (n,n) *a*, COMPLEX (KIND(0D0)), dimension (n) *p*, INTEGER, intent(in) *n* )

## 4.71 make\_noise.f90 File Reference

### Data Types

- module [Make\\_noise](#)

## 4.72 mdian1.f90 File Reference

### Functions/Subroutines

- subroutine [MDIAN1](#) (X2, N, XMED)

### 4.72.1 Function/Subroutine Documentation

4.72.1.1 subroutine **MDIAN1** ( `REAL(KIND(OD0))`, `dimension(n)` *X2*, `INTEGER` *N*,  
`REAL(KIND(OD0))` *XMED* )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.73 modelmod.f90 File Reference

### Data Types

- module [modelmod](#)

## 4.74 my\_git\_version.h File Reference

## 4.75 ompmod.f90 File Reference

### Data Types

- module [ompmod](#)

## 4.76 parfit.f90 File Reference

### Functions/Subroutines

- subroutine [parfit](#) (*fa*, *fb*, *fc*, *fmin*, *smin*)

### 4.76.1 Function/Subroutine Documentation

4.76.1.1 subroutine [parfit](#) ( *REAL* (KIND(0D0)) *fa*, *REAL* (KIND(0D0)) *fb*, *REAL* (KIND(0D0)) *fc*,  
*REAL* (KIND(0D0)) *fmin*, *REAL* (KIND(0D0)) *smin* )

Here is the caller graph for this function:



## 4.77 pathmod.f90 File Reference

### Data Types

- module [pathmod](#)

## 4.78 potana.f90 File Reference

### Functions/Subroutines

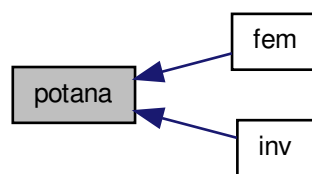
- subroutine [potana](#) (*l*, *k*, *my\_pota*)

### 4.78.1 Function/Subroutine Documentation



4.78.1.1 subroutine `potana` ( INTEGER (KIND=4) *l*, INTEGER (KIND=4) *k*, COMPLEX (KIND(OD0)), dimension(*sanz*) *my\_pota* )

Here is the caller graph for this function:



## 4.79 precal.f90 File Reference

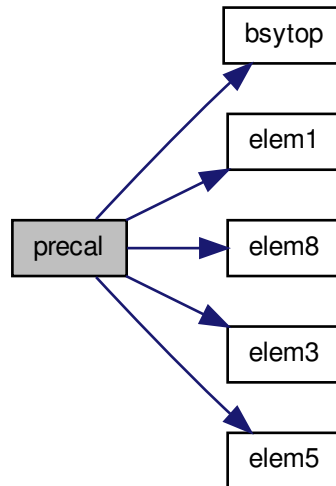
### Functions/Subroutines

- subroutine `prec`al ()

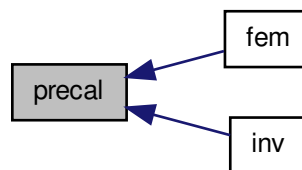
### 4.79.1 Function/Subroutine Documentation

#### 4.79.1.1 subroutine precal ( )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.80 rall.f90 File Reference

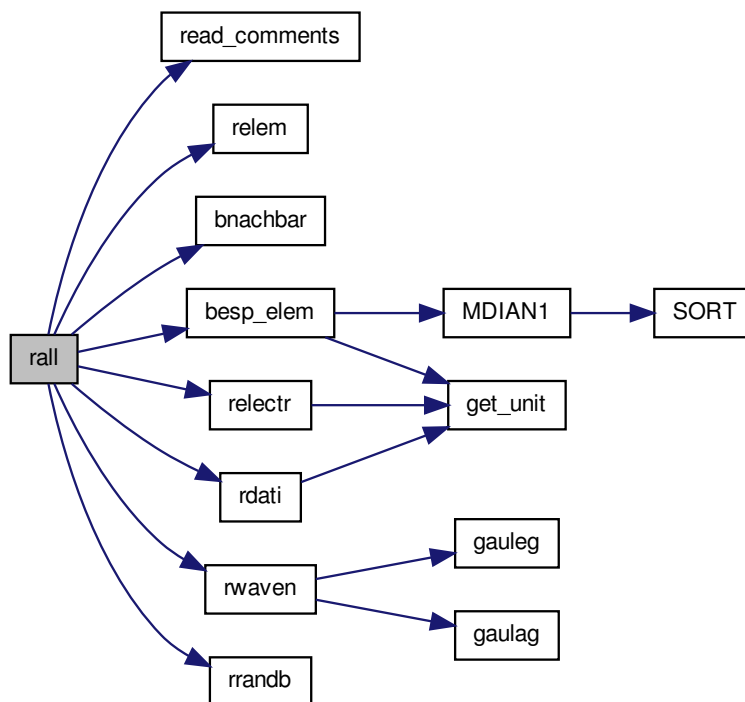
## Functions/Subroutines

- subroutine [rall](#) (kanal, delem, delectr, dstrom, drandb,

### 4.80.1 Function/Subroutine Documentation

4.80.1.1 subroutine `rall` ( `INTEGER (KIND = 4)` *kanal*, `CHARACTER (80)` *delem*, `CHARACTER (80)` *delectr*, `CHARACTER (80)` *dstrom*, `drandb` )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.81 randb.f90 File Reference

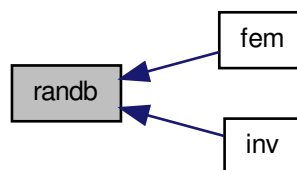
### Functions/Subroutines

- subroutine `randb` (`my_a`, `my_b`)

### 4.81.1 Function/Subroutine Documentation

4.81.1.1 subroutine `randb` ( `COMPLEX (KIND (0D0))`, dimension ((`mb+1`)\*`sanz`) `my_a`,  
`COMPLEX (KIND (0D0))`, dimension (`sanz`) `my_b` )

Here is the caller graph for this function:



## 4.82 randb2.f90 File Reference

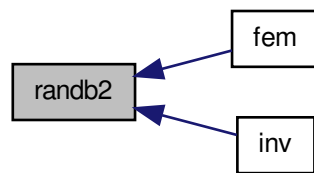
### Functions/Subroutines

- subroutine `randb2` (`my_a`, `my_b`)

### 4.82.1 Function/Subroutine Documentation

4.82.1.1 subroutine randb2 ( COMPLEX (KIND (0D0)), dimension ((mb+1)\*sanz) *my\_a*,  
COMPLEX (KIND (0D0)), dimension (sanz) *my\_b* )

Here is the caller graph for this function:



## 4.83 randbdc2.f90 File Reference

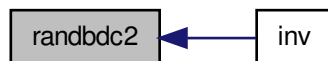
### Functions/Subroutines

- subroutine `randbdc2` (*my\_a*, *my\_b*)

### 4.83.1 Function/Subroutine Documentation

4.83.1.1 subroutine randbdc2 ( REAL (KIND (0D0)), dimension ((mb+1)\*sanz) *my\_a*, REAL  
(KIND (0D0)), dimension (sanz) *my\_b* )

Here is the caller graph for this function:



## 4.84 randbmod.f90 File Reference

### Data Types

- module [randbmod](#)

## 4.85 randdc.f90 File Reference

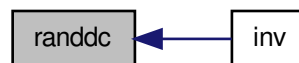
### Functions/Subroutines

- subroutine [randdc](#) (my\_a, my\_b)

### 4.85.1 Function/Subroutine Documentation

4.85.1.1 subroutine [randdc](#) ( REAL (KIND (0D0)), dimension ((mb+1)\*sanz) *my\_a*, REAL (KIND (0D0)), dimension (sanz) *my\_b* )

Here is the caller graph for this function:



## 4.86 rdati.f90 File Reference

### Functions/Subroutines

- subroutine [rdati](#) (kanal, datei)

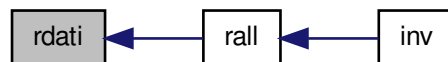
### 4.86.1 Function/Subroutine Documentation

4.86.1.1 subroutine `rdati` ( `INTEGER (KIND = 4)` *kanal*, `CHARACTER (80)` *datei* )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.87 rdatm.f90 File Reference

### Functions/Subroutines

- subroutine [rdatm](#) (`kanal`, `datei`)

### 4.87.1 Function/Subroutine Documentation

#### 4.87.1.1 subroutine `rdatm` ( `INTEGER(KIND = 4) kanal`, `CHARACTER(80) datei` )

Here is the caller graph for this function:



### 4.88 `refsig.f90` File Reference

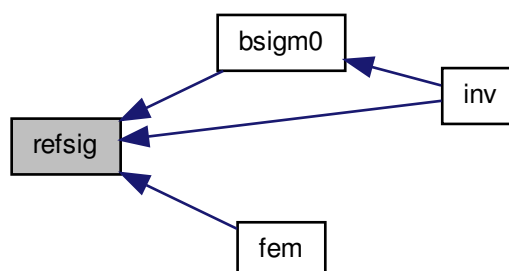
#### Functions/Subroutines

- subroutine `refsig` ( )

#### 4.88.1 Function/Subroutine Documentation

##### 4.88.1.1 subroutine `refsig` ( )

Here is the caller graph for this function:



### 4.89 `relectr.f90` File Reference



## Functions/Subroutines

- subroutine [relectr](#) (kanal, datei)

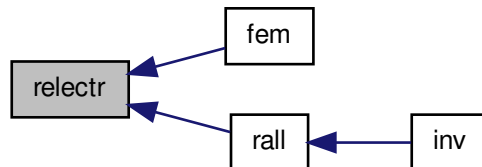
### 4.89.1 Function/Subroutine Documentation

4.89.1.1 subroutine [relectr](#) ( INTEGER (KIND = 4) *kanal*, CHARACTER (80) *datei* )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.90 relem.f90 File Reference

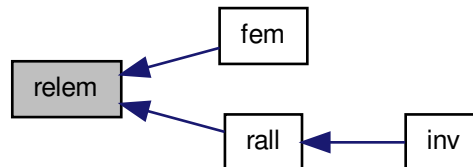
### Functions/Subroutines

- subroutine [relem](#) (kanal, datei)

### 4.90.1 Function/Subroutine Documentation

#### 4.90.1.1 subroutine relem ( INTEGER (KIND = 4) *kanal*, CHARACTER (80) *datei* )

Here is the caller graph for this function:



### 4.91 rrandb.f90 File Reference

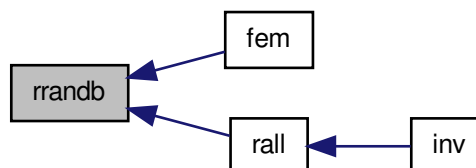
#### Functions/Subroutines

- subroutine `rrandb` (`kanal`, `datei`)

#### 4.91.1 Function/Subroutine Documentation

##### 4.91.1.1 subroutine rrandb ( INTEGER (KIND =4) *kanal*, CHARACTER (80) *datei* )

Here is the caller graph for this function:



### 4.92 rsigma.f90 File Reference

## Functions/Subroutines

- subroutine [rsigma](#) (kanal, datei)
- INTEGER function [set\\_ind\\_ref\\_grad](#) (i)
- subroutine [set\\_ind\\_ref\\_grad2](#)

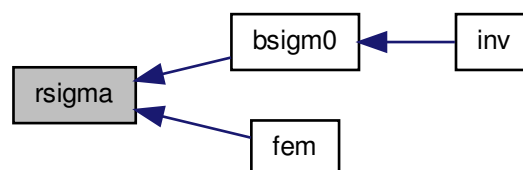
### 4.92.1 Function/Subroutine Documentation

4.92.1.1 subroutine [rsigma](#) ( INTEGER (KIND=4) *kanal*, CHARACTER (80) *datei* )

Here is the call graph for this function:



Here is the caller graph for this function:



4.92.1.2 INTEGER function [set\\_ind\\_ref\\_grad](#) ( INTEGER, intent(in) *i* )

#### 4.92.1.3 subroutine `set_ind_ref_grad2` ( )

Here is the call graph for this function:



### 4.93 `rtrafo.f90` File Reference

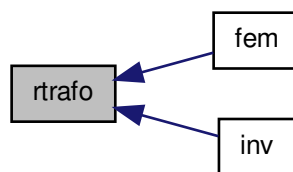
#### Functions/Subroutines

- subroutine `rtrafo` ( )

#### 4.93.1 Function/Subroutine Documentation

##### 4.93.1.1 subroutine `rtrafo` ( )

Here is the caller graph for this function:



### 4.94 `rwaven.f90` File Reference

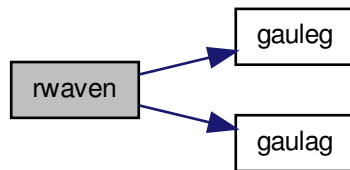
#### Functions/Subroutines

- subroutine `rwaven` ( )

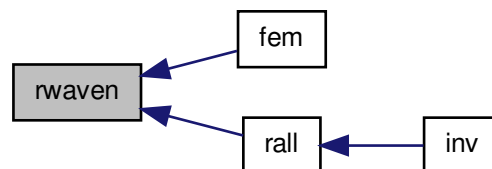
### 4.94.1 Function/Subroutine Documentation

#### 4.94.1.1 subroutine `rwaven` ( )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.95 scalab.f90 File Reference

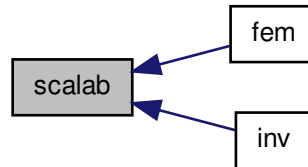
### Functions/Subroutines

- subroutine [scalab](#) (`a_scal`, `b_scal`, `fak_scal`)

### 4.95.1 Function/Subroutine Documentation

4.95.1.1 subroutine **scalab** ( COMPLEX (KIND(0D0)), dimension(\*) *a\_scal*, COMPLEX (KIND(0D0)), dimension(\*) *b\_scal*, REAL (KIND(0D0)), dimension(\*) *fak\_scal* )

Here is the caller graph for this function:



## 4.96 scaldc.f90 File Reference

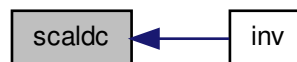
### Functions/Subroutines

- subroutine [scaldc](#) (*a\_scal*, *b\_scal*, *fak\_scal*)

### 4.96.1 Function/Subroutine Documentation

4.96.1.1 subroutine **scaldc** ( REAL(KIND(0D0)), dimension((mb+1)\*sanz) *a\_scal*, REAL(KIND(0D0)), dimension(sanz) *b\_scal*, REAL(KIND(0D0)), dimension(sanz) *fak\_scal* )

Here is the caller graph for this function:



## 4.97 semi-variogram.f90 File Reference

### Functions/Subroutines

- program [semi\\_variogram](#)

#### 4.97.1 Function/Subroutine Documentation

4.97.1.1 program [semi\\_variogram](#) ( )

### 4.98 sigmamod.f90 File Reference

#### Data Types

- module [sigmamod](#)

### 4.99 sort.f90 File Reference

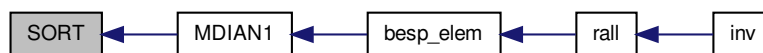
#### Functions/Subroutines

- subroutine [SORT](#) (N, RA)

#### 4.99.1 Function/Subroutine Documentation

4.99.1.1 subroutine [SORT](#) ( INTEGER *N*, REAL(KIND(0D0)), dimension(n) *RA* )

Here is the caller graph for this function:



### 4.100 tic\_toc.f90 File Reference

#### Data Types

- module [tic\\_toc](#)

## 4.101 update.f90 File Reference

### Functions/Subroutines

- subroutine `update` ()

*Unterprogramm zum Bestimmen und Anbringen der Modellverbesserung mittels 'Smoothness Least Squares Method' und konjugierten Gradienten.*

### 4.101.1 Function/Subroutine Documentation

#### 4.101.1.1 subroutine `update` ( )

Unterprogramm zum Bestimmen und Anbringen der Modellverbesserung mittels 'Smoothness Least Squares Method' und konjugierten Gradienten.

#### Author

Andreas Kemna

#### Date

01-Mar-1996

#### Author

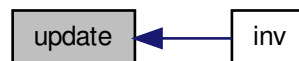
Roland Martin

#### Date

03-Aug-2009 until Sep. 2013

- translation to Fortran 90
- added general in code docu for specific parts
- added and tested triangular regularization (2009)
- added and tested stochastic regularization (2010-2011)
- added and tested reference model regularization (2012)
- added doxy comments (2013)

Here is the caller graph for this function:





## 4.102 variomodel.f90 File Reference

### Data Types

- module [variomodel](#)

## 4.103 vre.f90 File Reference

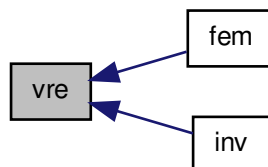
### Functions/Subroutines

- subroutine [vre](#) (a\_vre, b\_vre, pot\_vre)

#### 4.103.1 Function/Subroutine Documentation

4.103.1.1 subroutine [vre](#) ( COMPLEX (KIND(0D0)), dimension(\*) *a\_vre*, COMPLEX (KIND(0D0)), dimension(\*) *b\_vre*, COMPLEX (KIND(0D0)), dimension(\*) *pot\_vre* )

Here is the caller graph for this function:



## 4.104 vredc.f90 File Reference

### Functions/Subroutines

- subroutine [vredc](#) (a\_vre, b\_vre, pot\_vre)

#### 4.104.1 Function/Subroutine Documentation

4.104.1.1 subroutine `vredc` ( `REAL(KIND(0D0))`, `dimension((mb+1)*sanz)` *a\_vre*,  
`REAL(KIND(0D0))`, `dimension(sanz)` *b\_vre*, `COMPLEX(KIND(0D0))`, `dimension(sanz)`  
*pot\_vre* )

Here is the caller graph for this function:



## 4.105 wavenmod.f90 File Reference

### Data Types

- module [wavenmod](#)

## 4.106 wdatm.f90 File Reference

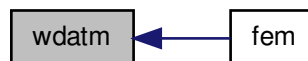
### Functions/Subroutines

- subroutine [wdatm](#) (*kanal*, *datei*)

### 4.106.1 Function/Subroutine Documentation

4.106.1.1 subroutine `wdatm` ( `INTEGER (KIND=4)` *kanal*, `CHARACTER (80)` *datei* )

Here is the caller graph for this function:



## 4.107 wkpot.f90 File Reference

### Functions/Subroutines

- subroutine [wkpot](#) (kanal, datei)

#### 4.107.1 Function/Subroutine Documentation

4.107.1.1 subroutine [wkpot](#) ( INTEGER (KIND=4) *kanal*, CHARACTER (80) *datei* )

Here is the caller graph for this function:



## 4.108 wout.f90 File Reference

### Functions/Subroutines

- subroutine [wout](#) (kanal, dsigma, dvolt)

#### 4.108.1 Function/Subroutine Documentation

4.108.1.1 subroutine [wout](#) ( INTEGER (KIND=4) *kanal*, CHARACTER (80) *dsigma*, CHARACTER (80) *dvolt* )

Here is the caller graph for this function:



## 4.109 wout\_up.f90 File Reference

### Functions/Subroutines

- subroutine `wout_up` (kanal, it, itr, switch)

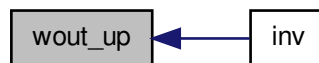
#### 4.109.1 Function/Subroutine Documentation

4.109.1.1 subroutine `wout_up` ( `INTEGER (KIND=4)`, intent(in) *kanal*, `INTEGER (KIND=4)`, intent(in) *it*, `INTEGER (KIND=4)`, intent(in) *itr*, `LOGICAL`, intent(in) *switch* )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.110 wpot.f90 File Reference

### Functions/Subroutines

- subroutine `wpot` (datei, np, mypot)

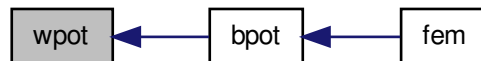
#### 4.110.1 Function/Subroutine Documentation

4.110.1.1 subroutine **wpot** ( CHARACTER (80) *datei*, INTEGER (KIND=4) *np*, COMPLEX (KIND(0D0)), dimension(*sanz*) *mypot* )

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.111 wsens.f90 File Reference

### Functions/Subroutines

- subroutine [wsens](#) (*kanal*, *datei*)

### 4.111.1 Function/Subroutine Documentation

4.111.1.1 subroutine `wsens` ( INTEGER (KIND=4) *kanal*, CHARACTER (80) *datei* )

Here is the caller graph for this function:

