

VMEC

8.52

Generated by Doxygen 1.9.1

Wed Apr 28 2021 15:44:00

1 Educational VMEC	1
1.1 Building	1
1.2 Example Execution	1
1.3 External NESTOR	2
1.4 Angle Constraint	2
2 Modules Index	3
2.1 Modules List	3
3 Data Type Index	5
3.1 Data Types List	5
4 File Index	7
4.1 File List	7
5 Module Documentation	11
5.1 fbal Module Reference	11
5.1.1 Variable Documentation	11
5.1.1.1 frcc_fac	11
5.1.1.2 fzsc_fac	11
5.1.1.3 rru_fac	12
5.1.1.4 rzu_fac	12
5.2 functions Module Reference	12
5.2.1 Function/Subroutine Documentation	12
5.2.1.1 function_test()	12
5.2.1.2 two_power()	13
5.2.1.3 two_power_gs()	14
5.3 line_segment Module Reference	15
5.3.1 Detailed Description	15
5.3.2 Function/Subroutine Documentation	16
5.3.2.1 line_seg()	16
5.3.2.2 line_seg_int()	16
5.3.2.3 line_seg_test()	17
5.4 mgrid_mod Module Reference	17
5.4.1 Detailed Description	19
5.4.2 Function/Subroutine Documentation	19
5.4.2.1 assign_bptrs()	19
5.4.2.2 free_mgrid()	20
5.4.2.3 read_mgrid()	20
5.4.2.4 sum_bfield()	21
5.4.3 Variable Documentation	22
5.4.3.1 abcoil	22
5.4.3.2 b_chi	22
5.4.3.3 bcoil	22

5.4.3.4 bloopnames	22
5.4.3.5 bpvac	22
5.4.3.6 brvac	23
5.4.3.7 bvac	23
5.4.3.8 bzvac	23
5.4.3.9 condif	23
5.4.3.10 curlabel	23
5.4.3.11 dbcoil	24
5.4.3.12 delrb	24
5.4.3.13 delzb	24
5.4.3.14 dsiext	24
5.4.3.15 dsilabel	24
5.4.3.16 iconnect	25
5.4.3.17 limitr	25
5.4.3.18 ln_next	25
5.4.3.19 mgrid_mode	25
5.4.3.20 mgrid_path	25
5.4.3.21 mgrid_path_old	26
5.4.3.22 nbcoil_max	26
5.4.3.23 nbcoils	26
5.4.3.24 nbcoilsn	26
5.4.3.25 nbfldn	26
5.4.3.26 nbsets	26
5.4.3.27 nbvac	27
5.4.3.28 needbfld	27
5.4.3.29 needflx	27
5.4.3.30 nextcur	27
5.4.3.31 nfper0	27
5.4.3.32 nlim	28
5.4.3.33 nlim_max	28
5.4.3.34 nlimset	28
5.4.3.35 nobd	28
5.4.3.36 nobser	28
5.4.3.37 np0b	28
5.4.3.38 nr0b	29
5.4.3.39 nrgrid	29
5.4.3.40 nsets	29
5.4.3.41 nsetsn	29
5.4.3.42 nz0b	29
5.4.3.43 nzgrid	29
5.4.3.44 pfcspec	30
5.4.3.45 plbfld	30

5.4.3.46 plflux	30
5.4.3.47 psiext	30
5.4.3.48 raw_coil_current	30
5.4.3.49 rbcoil	31
5.4.3.50 rbcoilsqr	31
5.4.3.51 reslim	31
5.4.3.52 rlim	31
5.4.3.53 rmaxb	31
5.4.3.54 rminb	32
5.4.3.55 rx1	32
5.4.3.56 rx2	32
5.4.3.57 seplim	32
5.4.3.58 tokid	32
5.4.3.59 unpsiext	32
5.4.3.60 vn_bp0	33
5.4.3.61 vn_br0	33
5.4.3.62 vn_bz0	33
5.4.3.63 vn_coilcur	33
5.4.3.64 vn_coilgrp	33
5.4.3.65 vn_ir	34
5.4.3.66 vn_jz	34
5.4.3.67 vn_kp	34
5.4.3.68 vn_mgmode	34
5.4.3.69 vn_nextcur	34
5.4.3.70 vn_nfp	35
5.4.3.71 vn_rmax	35
5.4.3.72 vn_rmin	35
5.4.3.73 vn_zmax	35
5.4.3.74 vn_zmin	35
5.4.3.75 xobser	36
5.4.3.76 xobsqr	36
5.4.3.77 zbcoil	36
5.4.3.78 zlim	36
5.4.3.79 zmaxb	36
5.4.3.80 zminb	37
5.4.3.81 zobser	37
5.4.3.82 zy1	37
5.4.3.83 zy2	37
5.5 nestor_io Module Reference	37
5.5.1 Detailed Description	40
5.5.2 Function/Subroutine Documentation	40
5.5.2.1 read_nestor_inputs()	41

5.5.2.2 write_nestor_outputs()	41
5.5.3 Variable Documentation	42
5.5.3.1 amatsav_nestor	42
5.5.3.2 amatsavdim	42
5.5.3.3 bsubvvac_nestor	43
5.5.3.4 bvecsav_nestor	43
5.5.3.5 bvecsavdim	43
5.5.3.6 ctor	43
5.5.3.7 extcur	43
5.5.3.8 ier_flag	44
5.5.3.9 input_extension	44
5.5.3.10 ivac	44
5.5.3.11 ivacskip	44
5.5.3.12 lasym	44
5.5.3.13 mgrid_file	45
5.5.3.14 mn1dim	45
5.5.3.15 mnmax	45
5.5.3.16 mnpd2_nestor	45
5.5.3.17 mnpotdim	45
5.5.3.18 mpol	46
5.5.3.19 nextcur	46
5.5.3.20 nextcurim	46
5.5.3.21 nfp	46
5.5.3.22 ntheta	46
5.5.3.23 ntor	47
5.5.3.24 nzeta	47
5.5.3.25 nzetadim	47
5.5.3.26 nzntdim	47
5.5.3.27 r2dim	47
5.5.3.28 raxis	48
5.5.3.29 rbtor	48
5.5.3.30 rmnc	48
5.5.3.31 rmns	48
5.5.3.32 signgs	48
5.5.3.33 vacuum_calls	49
5.5.3.34 vn_adm	49
5.5.3.35 vn_adp	49
5.5.3.36 vn_all_slm	49
5.5.3.37 vn_all_slp	49
5.5.3.38 vn_all_tlm	50
5.5.3.39 vn_all_tlp	50
5.5.3.40 vn_amatrix	50

5.5.3.41 vn_amatsav	50
5.5.3.42 vn_auu	50
5.5.3.43 vn_auv	51
5.5.3.44 vn_avv	51
5.5.3.45 vn_azm1u	51
5.5.3.46 vn_azp1u	51
5.5.3.47 vn_bexn	51
5.5.3.48 vn_bexni	52
5.5.3.49 vn_bexu	52
5.5.3.50 vn_bexv	52
5.5.3.51 vn_bphi	52
5.5.3.52 vn_bphiv	52
5.5.3.53 vn_brad	53
5.5.3.54 vn_brv	53
5.5.3.55 vn_bsqvac	53
5.5.3.56 vn_bsubu	53
5.5.3.57 vn_bsubv	53
5.5.3.58 vn_bsubvvac	54
5.5.3.59 vn_bvecsav	54
5.5.3.60 vn_bz	54
5.5.3.61 vn_bzv	54
5.5.3.62 vn_cma	54
5.5.3.63 vn_cma11u	55
5.5.3.64 vn_ctor	55
5.5.3.65 vn_delt1u	55
5.5.3.66 vn_drv	55
5.5.3.67 vn_extcur	55
5.5.3.68 vn_green	56
5.5.3.69 vn_greenp	56
5.5.3.70 vn_grpmn	56
5.5.3.71 vn_grpmn_m_map	56
5.5.3.72 vn_grpmn_n_map	56
5.5.3.73 vn_gstore	57
5.5.3.74 vn_guu_b	57
5.5.3.75 vn_guv_b	57
5.5.3.76 vn_gvv_b	57
5.5.3.77 vn_ier_flag	57
5.5.3.78 vn_imirr	58
5.5.3.79 vn_inputext	58
5.5.3.80 vn_ivac	58
5.5.3.81 vn_ivacskip	58
5.5.3.82 vn_lasym	58

5.5.3.83 vn_m_map	59
5.5.3.84 vn_mgrid	59
5.5.3.85 vn_mnmax	59
5.5.3.86 vn_mnpd	59
5.5.3.87 vn_mnpd2	59
5.5.3.88 vn_mpol	60
5.5.3.89 vn_n_map	60
5.5.3.90 vn_nfp	60
5.5.3.91 vn_ntheta	60
5.5.3.92 vn_ntor	60
5.5.3.93 vn_nzeta	61
5.5.3.94 vn_pmod	61
5.5.3.95 vn_potu	61
5.5.3.96 vn_potv	61
5.5.3.97 vn_potvac	61
5.5.3.98 vn_r0m	62
5.5.3.99 vn_r0p	62
5.5.3.100 vn_r1b	62
5.5.3.101 vn_r1m	62
5.5.3.102 vn_r1p	62
5.5.3.103 vn_ra1m	63
5.5.3.104 vn_ra1p	63
5.5.3.105 vn_raxis_nestor	63
5.5.3.106 vn_rbtor	63
5.5.3.107 vn_rcosuv	63
5.5.3.108 vn_rmnc	64
5.5.3.109 vn_rmns	64
5.5.3.110 vn_rsinuv	64
5.5.3.111 vn_rub	64
5.5.3.112 vn_ruu	64
5.5.3.113 vn_ruv	65
5.5.3.114 vn_rvb	65
5.5.3.115 vn_rvv	65
5.5.3.116 vn_rzb2	65
5.5.3.117 vn_signgs	65
5.5.3.118 vn_snr	66
5.5.3.119 vn_snv	66
5.5.3.120 vn_snz	66
5.5.3.121 vn_sqad1u	66
5.5.3.122 vn_sqad2u	66
5.5.3.123 vn_sqrta	67
5.5.3.124 vn_sqrtc	67

5.5.3.125 vn_tanu	67
5.5.3.126 vn_tanv	67
5.5.3.127 vn_tmod	67
5.5.3.128 vn_vacuum_calls	68
5.5.3.129 vn_wint	68
5.5.3.130 vn_xmpot	68
5.5.3.131 vn_xnpot	68
5.5.3.132 vn_z1b	68
5.5.3.133 vn_zaxis_nestor	69
5.5.3.134 vn_zmnc	69
5.5.3.135 vn_zmns	69
5.5.3.136 vn_zub	69
5.5.3.137 vn_zuu	69
5.5.3.138 vn_zuv	70
5.5.3.139 vn_zvb	70
5.5.3.140 vn_zvv	70
5.5.3.141 wint	70
5.5.3.142 xm	70
5.5.3.143 xn	71
5.5.3.144 zaxis	71
5.5.3.145 zmnc	71
5.5.3.146 zmns	71
5.6 read_wout_mod Module Reference	71
5.6.1 Detailed Description	79
5.6.2 Function/Subroutine Documentation	79
5.6.2.1 compute_currents()	79
5.6.2.2 loadrzl()	80
5.6.2.3 read_wout_deallocate()	80
5.6.2.4 readw_and_open()	81
5.6.2.5 tosuvspace()	81
5.6.3 Variable Documentation	81
5.6.3.1 ac	81
5.6.3.2 ac_aux_f	81
5.6.3.3 ac_aux_s	82
5.6.3.4 ai	82
5.6.3.5 ai_aux_f	82
5.6.3.6 ai_aux_s	82
5.6.3.7 alfa	82
5.6.3.8 am	83
5.6.3.9 am_aux_f	83
5.6.3.10 am_aux_s	83
5.6.3.11 aminor	83

5.6.3.12 anglemse	83
5.6.3.13 aspect	84
5.6.3.14 b0	84
5.6.3.15 bbc	84
5.6.3.16 bcwgt	84
5.6.3.17 bdotgradv	84
5.6.3.18 beta_vol	85
5.6.3.19 betapol	85
5.6.3.20 betator	85
5.6.3.21 betatot	85
5.6.3.22 betaxis	85
5.6.3.23 bmnc	86
5.6.3.24 bmns	86
5.6.3.25 bsubsmnc	86
5.6.3.26 bsubsmns	86
5.6.3.27 bsubumnc	86
5.6.3.28 bsubumns	87
5.6.3.29 bsubvmnc	87
5.6.3.30 bsubvmns	87
5.6.3.31 bsupumnc	87
5.6.3.32 bsupumns	87
5.6.3.33 bsupvmnc	88
5.6.3.34 bsupvmns	88
5.6.3.35 buco	88
5.6.3.36 bvco	88
5.6.3.37 chi	88
5.6.3.38 chipf	89
5.6.3.39 curmid	89
5.6.3.40 currumnc	89
5.6.3.41 currumns	89
5.6.3.42 currvmnc	89
5.6.3.43 currvmns	90
5.6.3.44 datastark	90
5.6.3.45 datathom	90
5.6.3.46 dcurr	90
5.6.3.47 delphid	90
5.6.3.48 dgeod	90
5.6.3.49 dmerc	91
5.6.3.50 dshear	91
5.6.3.51 dsiobt	91
5.6.3.52 dwell	91
5.6.3.53 equipf	91

5.6.3.54 extcur	92
5.6.3.55 flmwgt	92
5.6.3.56 fsql	92
5.6.3.57 fsqr	92
5.6.3.58 fsqt	92
5.6.3.59 fsqz	93
5.6.3.60 ftolv	93
5.6.3.61 gamma	93
5.6.3.62 gmnc	93
5.6.3.63 gmns	93
5.6.3.64 iasym	94
5.6.3.65 ierr_vmec	94
5.6.3.66 imatch_phiedge	94
5.6.3.67 imse	94
5.6.3.68 input_extension	94
5.6.3.69 ionlarmor	95
5.6.3.70 iotaf	95
5.6.3.71 iotas	95
5.6.3.72 ipnodes	95
5.6.3.73 isingng	95
5.6.3.74 isnodes	96
5.6.3.75 itfsq	96
5.6.3.76 itor	96
5.6.3.77 itse	96
5.6.3.78 jcuru	96
5.6.3.79 jcurv	97
5.6.3.80 jdotb	97
5.6.3.81 lasym	97
5.6.3.82 lmnc	97
5.6.3.83 lmns	97
5.6.3.84 ln_abeta	98
5.6.3.85 ln_ac	98
5.6.3.86 ln_ac_aux_f	98
5.6.3.87 ln_ac_aux_s	98
5.6.3.88 ln_actit	98
5.6.3.89 ln_ai	99
5.6.3.90 ln_ai_aux_f	99
5.6.3.91 ln_ai_aux_s	99
5.6.3.92 ln_am	99
5.6.3.93 ln_am_aux_f	99
5.6.3.94 ln_am_aux_s	100
5.6.3.95 ln_amin	100

5.6.3.96 ln_aspect	100
5.6.3.97 ln_asym	100
5.6.3.98 ln_b0	100
5.6.3.99 ln_beta	101
5.6.3.100 ln_betah	101
5.6.3.101 ln_bgrv	101
5.6.3.102 ln_bmnc	101
5.6.3.103 ln_bmns	101
5.6.3.104 ln_bsubsmnc	102
5.6.3.105 ln_bsubsmns	102
5.6.3.106 ln_bsubumnc	102
5.6.3.107 ln_bsubumnc_sur	102
5.6.3.108 ln_bsubumns	102
5.6.3.109 ln_bsubumns_sur	103
5.6.3.110 ln_bsubvmnc	103
5.6.3.111 ln_bsubvmnc_sur	103
5.6.3.112 ln_bsubvmns	103
5.6.3.113 ln_bsubvmns_sur	103
5.6.3.114 ln_bsupumnc	104
5.6.3.115 ln_bsupumnc_sur	104
5.6.3.116 ln_bsupumns	104
5.6.3.117 ln_bsupumns_sur	104
5.6.3.118 ln_bsupvmnc	104
5.6.3.119 ln_bsupvmnc_sur	105
5.6.3.120 ln_bsupvmns	105
5.6.3.121 ln_bsupvmns_sur	105
5.6.3.122 ln_buco	105
5.6.3.123 ln_bvco	105
5.6.3.124 ln_chi	106
5.6.3.125 ln_chipf	106
5.6.3.126 ln_ctor	106
5.6.3.127 ln_curlab	106
5.6.3.128 ln_equip	106
5.6.3.129 ln_error	107
5.6.3.130 ln_extcur	107
5.6.3.131 ln_extension	107
5.6.3.132 ln_fp	107
5.6.3.133 ln_free	107
5.6.3.134 ln_fsq	108
5.6.3.135 ln_fsqli	108
5.6.3.136 ln_fsqr	108
5.6.3.137 ln_fsqz	108

5.6.3.138 ln_ftolv	108
5.6.3.139 ln_gam	108
5.6.3.140 ln_gmnc	109
5.6.3.141 ln_gmns	109
5.6.3.142 ln_iotaf	109
5.6.3.143 ln_iotah	109
5.6.3.144 ln_jcuru	109
5.6.3.145 ln_jcurv	110
5.6.3.146 ln_jdotb	110
5.6.3.147 ln_lar	110
5.6.3.148 ln_lmnc	110
5.6.3.149 ln_lmns	110
5.6.3.150 ln_magen	111
5.6.3.151 ln_mass	111
5.6.3.152 ln_maxit	111
5.6.3.153 ln_maxmod	111
5.6.3.154 ln_maxmod_nyq	111
5.6.3.155 ln_maxr	112
5.6.3.156 ln_maxz	112
5.6.3.157 ln_mcurr	112
5.6.3.158 ln_merc	112
5.6.3.159 ln_mgeo	112
5.6.3.160 ln_mgrid	113
5.6.3.161 ln_minr	113
5.6.3.162 ln_modb	113
5.6.3.163 ln_mse	113
5.6.3.164 ln_mshear	113
5.6.3.165 ln_mwell	114
5.6.3.166 ln_pbeta	114
5.6.3.167 ln_pcurr_type	114
5.6.3.168 ln_phi	114
5.6.3.169 ln_phip	114
5.6.3.170 ln_phipf	115
5.6.3.171 ln_piota_type	115
5.6.3.172 ln_pmass_type	115
5.6.3.173 ln_pmod	115
5.6.3.174 ln_pmod_nyq	115
5.6.3.175 ln_polmod	116
5.6.3.176 ln_potvac	116
5.6.3.177 ln_presf	116
5.6.3.178 ln_presh	116
5.6.3.179 ln_qfact	116

5.6.3.180 ln_racc	117
5.6.3.181 ln_racs	117
5.6.3.182 ln_radnod	117
5.6.3.183 ln_rbc	117
5.6.3.184 ln_rbs	117
5.6.3.185 ln_rbt0	118
5.6.3.186 ln_rbt1	118
5.6.3.187 ln_recon	118
5.6.3.188 ln_rmaj	118
5.6.3.189 ln_rmnc	118
5.6.3.190 ln_rmns	119
5.6.3.191 ln_sgs	119
5.6.3.192 ln_specw	119
5.6.3.193 ln_tbeta	119
5.6.3.194 ln_therm	119
5.6.3.195 ln_thom	120
5.6.3.196 ln_tmod	120
5.6.3.197 ln_tmod_nyq	120
5.6.3.198 ln_tormod	120
5.6.3.199 ln_version	120
5.6.3.200 ln_vol	121
5.6.3.201 ln_vp	121
5.6.3.202 ln_wdot	121
5.6.3.203 ln_zacc	121
5.6.3.204 ln_zacs	121
5.6.3.205 ln_zbc	122
5.6.3.206 ln_zbs	122
5.6.3.207 ln_zmnc	122
5.6.3.208 ln_zmns	122
5.6.3.209 lthreed	122
5.6.3.210 lwout_opened	123
5.6.3.211 machsq	123
5.6.3.212 mass	123
5.6.3.213 mgrid_file	123
5.6.3.214 mnmax	123
5.6.3.215 mnmax_nyq	124
5.6.3.216 mnyq	124
5.6.3.217 mpol	124
5.6.3.218 msewgt	124
5.6.3.219 nfp	124
5.6.3.220 niter	125
5.6.3.221 nnyq	125

5.6.3.222 ns	125
5.6.3.223 nstore_seq	125
5.6.3.224 ntmax	125
5.6.3.225 ntor	126
5.6.3.226 overr	126
5.6.3.227 pcurr_type	126
5.6.3.228 pfac	126
5.6.3.229 phi	126
5.6.3.230 phidiam	127
5.6.3.231 phip	127
5.6.3.232 phipf	127
5.6.3.233 piota_type	127
5.6.3.234 pknots	127
5.6.3.235 pmass_type	128
5.6.3.236 potvac	128
5.6.3.237 pres	128
5.6.3.238 presf	128
5.6.3.239 presmid	128
5.6.3.240 qfact	129
5.6.3.241 qmeas	129
5.6.3.242 qmid	129
5.6.3.243 raxis	129
5.6.3.244 rbtor	129
5.6.3.245 rbtor0	130
5.6.3.246 rmajor	130
5.6.3.247 rmax_surf	130
5.6.3.248 rmid	130
5.6.3.249 rmin_surf	130
5.6.3.250 rmnc	131
5.6.3.251 rmns	131
5.6.3.252 rstark	131
5.6.3.253 rthom	131
5.6.3.254 rzl_local	131
5.6.3.255 shear	132
5.6.3.256 sknots	132
5.6.3.257 specw	132
5.6.3.258 tswgt	132
5.6.3.259 version_	132
5.6.3.260 vn_abeta	132
5.6.3.261 vn_ac	133
5.6.3.262 vn_ac_aux_f	133
5.6.3.263 vn_ac_aux_s	133

5.6.3.264 vn_actit	133
5.6.3.265 vn_ah	133
5.6.3.266 vn_ai	134
5.6.3.267 vn_ai_aux_f	134
5.6.3.268 vn_ai_aux_s	134
5.6.3.269 vn_am	134
5.6.3.270 vn_am_aux_f	134
5.6.3.271 vn_am_aux_s	135
5.6.3.272 vn_amin	135
5.6.3.273 vn_aspect	135
5.6.3.274 vn_asym	135
5.6.3.275 vn_atuname	135
5.6.3.276 vn_b0	136
5.6.3.277 vn_beta	136
5.6.3.278 vn_betah	136
5.6.3.279 vn_bgrv	136
5.6.3.280 vn_bmnc	136
5.6.3.281 vn_bmns	137
5.6.3.282 vn_bsubsmnc	137
5.6.3.283 vn_bsubsmns	137
5.6.3.284 vn_bsubumnc	137
5.6.3.285 vn_bsubumnc_sur	137
5.6.3.286 vn_bsubumns	138
5.6.3.287 vn_bsubumns_sur	138
5.6.3.288 vn_bsubvmnc	138
5.6.3.289 vn_bsubvmnc_sur	138
5.6.3.290 vn_bsubvmns	138
5.6.3.291 vn_bsubvmns_sur	139
5.6.3.292 vn_bsupumnc	139
5.6.3.293 vn_bsupumnc_sur	139
5.6.3.294 vn_bsupumns	139
5.6.3.295 vn_bsupumns_sur	139
5.6.3.296 vn_bsupvmnc	139
5.6.3.297 vn_bsupvmnc_sur	140
5.6.3.298 vn_bsupvmns	140
5.6.3.299 vn_bsupvmns_sur	140
5.6.3.300 vn_buco	140
5.6.3.301 vn_bvco	140
5.6.3.302 vn_chi	141
5.6.3.303 vn_chipf	141
5.6.3.304 vn_ctor	141
5.6.3.305 vn_curlab	141

5.6.3.306 vn_equip	141
5.6.3.307 vn_error	142
5.6.3.308 vn_extcur	142
5.6.3.309 vn_extension	142
5.6.3.310 vn_fp	142
5.6.3.311 vn_free	142
5.6.3.312 vn_fsq	143
5.6.3.313 vn_fsqli	143
5.6.3.314 vn_fsqr	143
5.6.3.315 vn_fsqz	143
5.6.3.316 vn_ftolv	143
5.6.3.317 vn_gam	144
5.6.3.318 vn_gmnc	144
5.6.3.319 vn_gmns	144
5.6.3.320 vn_iotaf	144
5.6.3.321 vn_iotah	144
5.6.3.322 vn_jcuru	145
5.6.3.323 vn_jcurv	145
5.6.3.324 vn_jdotb	145
5.6.3.325 vn_lar	145
5.6.3.326 vn_lmnc	145
5.6.3.327 vn_lmns	146
5.6.3.328 vn_magen	146
5.6.3.329 vn_mass	146
5.6.3.330 vn_maxit	146
5.6.3.331 vn_maxmod	146
5.6.3.332 vn_maxmod_nyq	147
5.6.3.333 vn_maxr	147
5.6.3.334 vn_maxz	147
5.6.3.335 vn_mcurr	147
5.6.3.336 vn_merc	147
5.6.3.337 vn_mgeo	148
5.6.3.338 vn_mgrid	148
5.6.3.339 vn_minr	148
5.6.3.340 vn_modb	148
5.6.3.341 vn_mse	148
5.6.3.342 vn_mshear	149
5.6.3.343 vn_mwell	149
5.6.3.344 vn_overr	149
5.6.3.345 vn_pbeta	149
5.6.3.346 vn_pcurr_type	149
5.6.3.347 vn_phi	150

5.6.3.348 vn_phip	150
5.6.3.349 vn_phipf	150
5.6.3.350 vn_piota_type	150
5.6.3.351 vn_pmass_type	150
5.6.3.352 vn_pmod	151
5.6.3.353 vn_pmod_nyq	151
5.6.3.354 vn_polmod	151
5.6.3.355 vn_potvac	151
5.6.3.356 vn_presf	151
5.6.3.357 vn_presh	152
5.6.3.358 vn_qfact	152
5.6.3.359 vn_racc	152
5.6.3.360 vn_racs	152
5.6.3.361 vn_radnod	152
5.6.3.362 vn_rbc	153
5.6.3.363 vn_rbs	153
5.6.3.364 vn_rbt0	153
5.6.3.365 vn_rbt1	153
5.6.3.366 vn_rmaj	153
5.6.3.367 vn_rmnc	154
5.6.3.368 vn_rmns	154
5.6.3.369 vn_sgs	154
5.6.3.370 vn_specw	154
5.6.3.371 vn_tbeta	154
5.6.3.372 vn_therm	155
5.6.3.373 vn_thom	155
5.6.3.374 vn_tmod	155
5.6.3.375 vn_tmod_nyq	155
5.6.3.376 vn_tormod	155
5.6.3.377 vn_version	156
5.6.3.378 vn_vol	156
5.6.3.379 vn_vp	156
5.6.3.380 vn_wdot	156
5.6.3.381 vn_zacc	156
5.6.3.382 vn_zacs	157
5.6.3.383 vn_zbc	157
5.6.3.384 vn_zbs	157
5.6.3.385 vn_zmnc	157
5.6.3.386 vn_zmns	157
5.6.3.387 volavgb	158
5.6.3.388 volume	158
5.6.3.389 vp	158

5.6.3.390 wb	158
5.6.3.391 wdot	158
5.6.3.392 wp	159
5.6.3.393 xm	159
5.6.3.394 xm_nyq	159
5.6.3.395 xn	159
5.6.3.396 xn_nyq	159
5.6.3.397 y2stark	160
5.6.3.398 y2thom	160
5.6.3.399 ystark	160
5.6.3.400 ythom	160
5.6.3.401 zaxis	160
5.6.3.402 zmax_surf	160
5.6.3.403 zmnc	161
5.6.3.404 zmns	161
5.7 realspace Module Reference	161
5.7.1 Variable Documentation	162
5.7.1.1 chip	162
5.7.1.2 extra1	162
5.7.1.3 extra2	162
5.7.1.4 extra3	162
5.7.1.5 extra4	163
5.7.1.6 gcon	163
5.7.1.7 guu	163
5.7.1.8 guv	163
5.7.1.9 gvv	163
5.7.1.10 phip	164
5.7.1.11 r1	164
5.7.1.12 rcon	164
5.7.1.13 rcon0	164
5.7.1.14 ru	164
5.7.1.15 ru0	165
5.7.1.16 rv	165
5.7.1.17 shalf	165
5.7.1.18 sqrts	165
5.7.1.19 wint	166
5.7.1.20 z1	166
5.7.1.21 zcon	166
5.7.1.22 zcon0	166
5.7.1.23 zu	166
5.7.1.24 zu0	167
5.7.1.25 zv	167

5.8 <code>safe_open_mod</code> Module Reference	167
5.8.1 Function/Subroutine Documentation	167
5.8.1.1 <code>safe_open()</code>	167
5.9 <code>stel_constants</code> Module Reference	168
5.9.1 Variable Documentation	168
5.9.1.1 <code>degree</code>	168
5.9.1.2 <code>mu0</code>	168
5.9.1.3 <code>one</code>	168
5.9.1.4 <code>pi</code>	169
5.9.1.5 <code>pio2</code>	169
5.9.1.6 <code>sqrt2</code>	169
5.9.1.7 <code>twopi</code>	169
5.9.1.8 <code>zero</code>	169
5.10 <code>stel_kinds</code> Module Reference	169
5.10.1 Variable Documentation	170
5.10.1.1 <code>cprec</code>	170
5.10.1.2 <code>dp</code>	170
5.10.1.3 <code>iprec</code>	170
5.10.1.4 <code>rprec</code>	170
5.11 <code>vac_persistent</code> Module Reference	171
5.11.1 Variable Documentation	171
5.11.1.1 <code>bsubu_sur</code>	171
5.11.1.2 <code>bsubv_sur</code>	171
5.11.1.3 <code>bsupu_sur</code>	172
5.11.1.4 <code>bsupv_sur</code>	172
5.11.1.5 <code>cmns</code>	172
5.11.1.6 <code>cosper</code>	172
5.11.1.7 <code>cosu</code>	172
5.11.1.8 <code>cosu1</code>	173
5.11.1.9 <code>cosui</code>	173
5.11.1.10 <code>cosuv</code>	173
5.11.1.11 <code>cosv</code>	173
5.11.1.12 <code>cosv1</code>	173
5.11.1.13 <code>csign</code>	174
5.11.1.14 <code>imirr</code>	174
5.11.1.15 <code>sinper</code>	174
5.11.1.16 <code>sinu</code>	174
5.11.1.17 <code>sinu1</code>	174
5.11.1.18 <code>sinui</code>	175
5.11.1.19 <code>sinuv</code>	175
5.11.1.20 <code>sinv</code>	175
5.11.1.21 <code>sinv1</code>	175

5.11.1.22 tanu	175
5.11.1.23 tanu_1d	176
5.11.1.24 tanv	176
5.11.1.25 tanv_1d	176
5.11.1.26 xmpot	176
5.11.1.27 xnpot	176
5.12 vacmod Module Reference	177
5.12.1 Function/Subroutine Documentation	179
5.12.1.1 allocate_nestor()	179
5.12.1.2 free_mem_nestor()	180
5.12.2 Variable Documentation	180
5.12.2.1 actemp	180
5.12.2.2 adm	180
5.12.2.3 adp	181
5.12.2.4 all_slm	181
5.12.2.5 all_slp	181
5.12.2.6 all_tlm	181
5.12.2.7 all_tlp	181
5.12.2.8 alp	182
5.12.2.9 alu	182
5.12.2.10 alv	182
5.12.2.11 alvp	182
5.12.2.12 amatrix	182
5.12.2.13 amatsav	183
5.12.2.14 astemp	183
5.12.2.15 auu	183
5.12.2.16 auv	183
5.12.2.17 avv	183
5.12.2.18 azm1u	184
5.12.2.19 azp1u	184
5.12.2.20 bcos	184
5.12.2.21 bexn	184
5.12.2.22 bexni	184
5.12.2.23 bexu	185
5.12.2.24 bexv	185
5.12.2.25 bphi	185
5.12.2.26 bphiv	185
5.12.2.27 brad	185
5.12.2.28 brv	186
5.12.2.29 bsin	186
5.12.2.30 bsqvac	186
5.12.2.31 bsubu	186

5.12.2.32 bsubv	186
5.12.2.33 bsubvvac	187
5.12.2.34 bvecsav	187
5.12.2.35 bz	187
5.12.2.36 bzv	187
5.12.2.37 cma	187
5.12.2.38 cma11u	188
5.12.2.39 delt1u	188
5.12.2.40 drv	188
5.12.2.41 dsave	188
5.12.2.42 g1	188
5.12.2.43 g2	189
5.12.2.44 ga1	189
5.12.2.45 ga2	189
5.12.2.46 green	189
5.12.2.47 greenp	189
5.12.2.48 grpmn	190
5.12.2.49 grpmn_m_map_wrt	190
5.12.2.50 grpmn_n_map_wrt	190
5.12.2.51 gsave	190
5.12.2.52 gstore	190
5.12.2.53 guu_b	191
5.12.2.54 guv_b	191
5.12.2.55 gvv_b	191
5.12.2.56 m_map_wrt	191
5.12.2.57 n_map_wrt	191
5.12.2.58 onp	192
5.12.2.59 onp2	192
5.12.2.60 p5	192
5.12.2.61 pi2	192
5.12.2.62 pi3	192
5.12.2.63 pi4	193
5.12.2.64 potu	193
5.12.2.65 potv	193
5.12.2.66 potvac	193
5.12.2.67 precal_done	193
5.12.2.68 r0m	194
5.12.2.69 r0p	194
5.12.2.70 r1b	194
5.12.2.71 r1m	194
5.12.2.72 r1p	194
5.12.2.73 ra1m	195

5.12.2.74 ra1p	195
5.12.2.75 raxis_nestor	195
5.12.2.76 rcosuv	195
5.12.2.77 rsinuv	195
5.12.2.78 rub	196
5.12.2.79 ruu	196
5.12.2.80 ruv	196
5.12.2.81 rvb	196
5.12.2.82 rvv	196
5.12.2.83 rzb2	197
5.12.2.84 slm	197
5.12.2.85 slp	197
5.12.2.86 slpm	197
5.12.2.87 snr	197
5.12.2.88 snv	198
5.12.2.89 snz	198
5.12.2.90 source	198
5.12.2.91 squad1u	198
5.12.2.92 squad2u	198
5.12.2.93 sqrt a	199
5.12.2.94 sqrt c	199
5.12.2.95 tlm	199
5.12.2.96 tlm1	199
5.12.2.97 tlm2	199
5.12.2.98 tlp	200
5.12.2.99 tlp1	200
5.12.2.100 tlp2	200
5.12.2.101 tlp m	200
5.12.2.102 two	200
5.12.2.103 xpts	201
5.12.2.104 z1b	201
5.12.2.105 zaxis_nestor	201
5.12.2.106 zub	201
5.12.2.107 zuu	201
5.12.2.108 zuv	202
5.12.2.109 zvb	202
5.12.2.110 zvv	202
5.13 vacmod0 Module Reference	202
5.13.1 Function/Subroutine Documentation	203
5.13.1.1 set_nestor_sizes()	203
5.13.2 Variable Documentation	203
5.13.2.1 mf	203

5.13.2.2 mf1	204
5.13.2.3 mnpd	204
5.13.2.4 mnpd2	204
5.13.2.5 ndim	204
5.13.2.6 nf	204
5.13.2.7 nf1	205
5.13.2.8 nfper	205
5.13.2.9 nu	205
5.13.2.10 nu2	205
5.13.2.11 nu3	205
5.13.2.12 nuv	206
5.13.2.13 nuv2	206
5.13.2.14 nuv_tan	206
5.13.2.15 nv	206
5.13.2.16 nvp	206
5.13.2.17 nvper	207
5.14 vforces Module Reference	207
5.14.1 Variable Documentation	207
5.14.1.1 armn	207
5.14.1.2 armn_e	208
5.14.1.3 armn_o	208
5.14.1.4 azmn	208
5.14.1.5 azmn_e	208
5.14.1.6 azmn_o	208
5.14.1.7 blmn	209
5.14.1.8 blmn_e	209
5.14.1.9 blmn_o	209
5.14.1.10 brmn	209
5.14.1.11 brmn_e	209
5.14.1.12 brmn_o	210
5.14.1.13 bzmnn	210
5.14.1.14 bzmnn_e	210
5.14.1.15 bzmnn_o	210
5.14.1.16 clmn	210
5.14.1.17 clmn_e	211
5.14.1.18 clmn_o	211
5.14.1.19 crmn	211
5.14.1.20 crmn_e	211
5.14.1.21 crmn_o	211
5.14.1.22 czmn	212
5.14.1.23 czmn_e	212
5.14.1.24 czmn_o	212

5.15 vmec_dim Module Reference	212
5.15.1 Variable Documentation	212
5.15.1.1 mnmax	213
5.15.1.2 mnmax_nyq	213
5.15.1.3 mns	213
5.15.1.4 mnsiz	213
5.15.1.5 mpol1	213
5.15.1.6 nrzt	213
5.15.1.7 ns	214
5.15.1.8 ns1	214
5.15.1.9 ns_maxval	214
5.15.1.10 ntheta1	214
5.15.1.11 ntheta2	214
5.15.1.12 ntheta3	215
5.15.1.13 ntor1	215
5.15.1.14 nznt	215
5.16 vmec_input Module Reference	215
5.16.1 Function/Subroutine Documentation	216
5.16.1.1 read_indata_namelist()	217
5.16.1.2 write_indata_namelist()	217
5.16.2 Variable Documentation	217
5.16.2.1 ac	217
5.16.2.2 ac_aux_f	218
5.16.2.3 ac_aux_s	218
5.16.2.4 ai	218
5.16.2.5 ai_aux_f	218
5.16.2.6 ai_aux_s	218
5.16.2.7 am	219
5.16.2.8 am_aux_f	219
5.16.2.9 am_aux_s	219
5.16.2.10 aphi	219
5.16.2.11 bloat	219
5.16.2.12 curt	220
5.16.2.13 delt	220
5.16.2.14 extcur	220
5.16.2.15 ftol_array	220
5.16.2.16 ftol_default	220
5.16.2.17 gamma	221
5.16.2.18 input_extension	221
5.16.2.19 lasym	221
5.16.2.20 lbsubs	221
5.16.2.21 lfreeb	221

5.16.2.22 mfilter_fbdy	222
5.16.2.23 mgrid_file	222
5.16.2.24 mpol	222
5.16.2.25 mpol_default	222
5.16.2.26 ncurr	222
5.16.2.27 nfilter_fbdy	223
5.16.2.28 nfp	223
5.16.2.29 niter_array	223
5.16.2.30 niter_default	223
5.16.2.31 ns_array	223
5.16.2.32 ns_default	224
5.16.2.33 nstep	224
5.16.2.34 ntheta	224
5.16.2.35 ntor	224
5.16.2.36 ntor_default	224
5.16.2.37 nvacskip	225
5.16.2.38 nzeta	225
5.16.2.39 pcurr_type	225
5.16.2.40 phiedge	225
5.16.2.41 piota_type	225
5.16.2.42 pmass_type	226
5.16.2.43 pres_scale	226
5.16.2.44 raxis_cc	226
5.16.2.45 raxis_cs	226
5.16.2.46 rbc	226
5.16.2.47 rbs	227
5.16.2.48 spres_ped	227
5.16.2.49 tcon0	227
5.16.2.50 zaxis_cc	227
5.16.2.51 zaxis_cs	227
5.16.2.52 zbc	228
5.16.2.53 zbs	228
5.17 vmec_io Module Reference	228
5.17.1 Variable Documentation	228
5.17.1.1 aminor_p	228
5.17.1.2 b0	229
5.17.1.3 betapol	229
5.17.1.4 betator	229
5.17.1.5 betatot	229
5.17.1.6 betaxis	229
5.17.1.7 circum_p	230
5.17.1.8 cross_area_p	230

5.17.1.9 ionlarmor	230
5.17.1.10 kappa_p	230
5.17.1.11 rmajor_p	230
5.17.1.12 rmax_surf	231
5.17.1.13 rmin_surf	231
5.17.1.14 surf_area_p	231
5.17.1.15 volavgb	231
5.17.1.16 volume_p	231
5.17.1.17 zmax_surf	232
5.18 vmec_main Module Reference	232
5.18.1 Variable Documentation	235
5.18.1.1 ard	235
5.18.1.2 arm	235
5.18.1.3 aspect	235
5.18.1.4 azd	235
5.18.1.5 azm	235
5.18.1.6 bdamp	236
5.18.1.7 bdotb	236
5.18.1.8 bdotgradv	236
5.18.1.9 beta_vol	236
5.18.1.10 blam	236
5.18.1.11 bmax	237
5.18.1.12 bmin	237
5.18.1.13 bpedge	237
5.18.1.14 brd	237
5.18.1.15 bredge	237
5.18.1.16 brm	238
5.18.1.17 bsqsav	238
5.18.1.18 bsubu0	238
5.18.1.19 buco	238
5.18.1.20 bucof	238
5.18.1.21 bvco	239
5.18.1.22 bvcof	239
5.18.1.23 bzd	239
5.18.1.24 bzedge	239
5.18.1.25 bzm	239
5.18.1.26 chi	240
5.18.1.27 chipf	240
5.18.1.28 chips	240
5.18.1.29 clam	240
5.18.1.30 crd	240
5.18.1.31 ctor	241

5.18.1.32 currv	241
5.18.1.33 dbsq	241
5.18.1.34 delbsq	241
5.18.1.35 delt0r	241
5.18.1.36 dlam	242
5.18.1.37 equip	242
5.18.1.38 faccon	242
5.18.1.39 faclam	242
5.18.1.40 faclam0	242
5.18.1.41 fedge	243
5.18.1.42 fnorm	243
5.18.1.43 fnorm1	243
5.18.1.44 fnorml	243
5.18.1.45 fsq	243
5.18.1.46 fsql	244
5.18.1.47 fsql1	244
5.18.1.48 fsqr	244
5.18.1.49 fsqr1	244
5.18.1.50 fsqsum0	244
5.18.1.51 fsqz	245
5.18.1.52 fsqz1	245
5.18.1.53 ftolv	245
5.18.1.54 hi	245
5.18.1.55 hp	245
5.18.1.56 hs	246
5.18.1.57 icurv	246
5.18.1.58 iequi	246
5.18.1.59 ijacob	246
5.18.1.60 iotaf	247
5.18.1.61 iotas	247
5.18.1.62 ireflect	247
5.18.1.63 irst	247
5.18.1.64 irzloff	248
5.18.1.65 iter1	248
5.18.1.66 iter2	248
5.18.1.67 itfsq	248
5.18.1.68 ivac	249
5.18.1.69 jcuru	249
5.18.1.70 jcurv	249
5.18.1.71 jdotb	249
5.18.1.72 jpar2	249
5.18.1.73 jperp2	250

5.18.1.74 lconm1	250
5.18.1.75 lflip	250
5.18.1.76 lthreed	250
5.18.1.77 mass	250
5.18.1.78 multi_ns_grid	251
5.18.1.79 ndatai	251
5.18.1.80 ndatap	251
5.18.1.81 neqs	251
5.18.1.82 niterv	251
5.18.1.83 ohs	252
5.18.1.84 otau	252
5.18.1.85 otav	252
5.18.1.86 overr	252
5.18.1.87 phi	252
5.18.1.88 phipf	253
5.18.1.89 phips	253
5.18.1.90 pres	253
5.18.1.91 presf	253
5.18.1.92 presgrad	253
5.18.1.93 psi	254
5.18.1.94 r00	254
5.18.1.95 r0scale	254
5.18.1.96 rbsq	254
5.18.1.97 rbtor	254
5.18.1.98 rbtor0	255
5.18.1.99 res0	255
5.18.1.100 rinner	255
5.18.1.101 rmn_bdy	255
5.18.1.102 router	255
5.18.1.103 sifa	256
5.18.1.104 sifa2	256
5.18.1.105 sm	256
5.18.1.106 sp	256
5.18.1.107 specw	256
5.18.1.108 spfa	257
5.18.1.109 spfa2	257
5.18.1.110 tcon	257
5.18.1.111 vacuum_calls	257
5.18.1.112 voli	257
5.18.1.113 vp	258
5.18.1.114 vpphi	258
5.18.1.115 wb	258

5.18.1.116 wp	258
5.18.1.117 xcl0	258
5.18.1.118 xmpq	259
5.18.1.119 yellip	259
5.18.1.120 ygeo	259
5.18.1.121 yinden	259
5.18.1.122 yshift	259
5.18.1.123 ytrian	260
5.18.1.124 z00	260
5.18.1.125 zmn_bdy	260
5.19 vmec_params Module Reference	260
5.19.1 Variable Documentation	261
5.19.1.1 bad_jacobian_flag	261
5.19.1.2 cleanup_flag	262
5.19.1.3 input_error_flag	262
5.19.1.4 jac75_flag	262
5.19.1.5 jlam	262
5.19.1.6 jmin1	262
5.19.1.7 jmin2	263
5.19.1.8 lamscale	263
5.19.1.9 m0	263
5.19.1.10 m1	263
5.19.1.11 meven	264
5.19.1.12 misc_error_flag	264
5.19.1.13 mnyq	264
5.19.1.14 modd	264
5.19.1.15 mscale	264
5.19.1.16 n0	265
5.19.1.17 ndamp	265
5.19.1.18 nnyq	265
5.19.1.19 norm_term_flag	265
5.19.1.20 ns4	265
5.19.1.21 ns_error_flag	266
5.19.1.22 nscale	266
5.19.1.23 ntmax	266
5.19.1.24 output_flag	266
5.19.1.25 pdamp	266
5.19.1.26 phiedge_error_flag	267
5.19.1.27 rcc	267
5.19.1.28 rcs	267
5.19.1.29 readin_flag	267
5.19.1.30 reset_jacdt_flag	267

5.19.1.31 restart_flag	268
5.19.1.32 rsc	268
5.19.1.33 rss	268
5.19.1.34 signgs	268
5.19.1.35 successful_term_flag	268
5.19.1.36 timestep_flag	269
5.19.1.37 uminus	269
5.19.1.38 version_	269
5.19.1.39 zcc	269
5.19.1.40 zcs	269
5.19.1.41 zsc	270
5.19.1.42 zss	270
5.20 vmec_persistent Module Reference	270
5.20.1 Variable Documentation	270
5.20.1.1 cos01	271
5.20.1.2 cosmu	271
5.20.1.3 cosmui	271
5.20.1.4 cosmui3	271
5.20.1.5 cosmum	271
5.20.1.6 cosmumi	271
5.20.1.7 cosmumi3	272
5.20.1.8 cosnv	272
5.20.1.9 cosnvn	272
5.20.1.10 ixm	272
5.20.1.11 jmin3	272
5.20.1.12 sin01	272
5.20.1.13 sinmu	273
5.20.1.14 sinmui	273
5.20.1.15 sinmum	273
5.20.1.16 sinmumi	273
5.20.1.17 sinnv	273
5.20.1.18 sinnvn	274
5.20.1.19 xm	274
5.20.1.20 xm_nyq	274
5.20.1.21 xn	274
5.20.1.22 xn_nyq	274
5.21 vmecier Module Reference	275
5.21.1 Variable Documentation	275
5.21.1.1 dcurr	275
5.21.1.2 dgeod	275
5.21.1.3 dmerc	275
5.21.1.4 dshear	276

5.21.1.5 dwell	276
5.22 vparams Module Reference	276
5.22.1 Variable Documentation	277
5.22.1.1 c1pm13	277
5.22.1.2 c1pm2	277
5.22.1.3 c1pm8	277
5.22.1.4 c2p0	277
5.22.1.5 c3p0	277
5.22.1.6 cbig	278
5.22.1.7 cp05	278
5.22.1.8 cp15	278
5.22.1.9 cp25	278
5.22.1.10 cp5	278
5.22.1.11 indata0	278
5.22.1.12 jxbout0	279
5.22.1.13 mpol1d	279
5.22.1.14 mpold	279
5.22.1.15 ndatafmax	279
5.22.1.16 nfort18	279
5.22.1.17 nmercier0	279
5.22.1.18 nsd	280
5.22.1.19 nstore_seq	280
5.22.1.20 nthreed	280
5.22.1.21 nthreed0	280
5.22.1.22 ntor1d	280
5.22.1.23 ntord	281
5.22.1.24 nwout0	281
5.22.1.25 osqrt2	281
5.23 vsvd0 Module Reference	281
5.23.1 Variable Documentation	281
5.23.1.1 nigroup	281
5.24 xstuff Module Reference	282
5.24.1 Variable Documentation	282
5.24.1.1 gc	282
5.24.1.2 scalxc	282
5.24.1.3 xc	282
5.24.1.4 xcdot	283
5.24.1.5 xsave	283
5.24.1.6 xstore	283
6 Data Type Documentation	285
6.1 read_wout_mod::read_wout_file Interface Reference	285

6.1.1 Detailed Description	285
6.1.2 Member Function/Subroutine Documentation	285
6.1.2.1 readw_and_open()	285
7 File Documentation	289
7.1 README.md File Reference	289
7.2 src/add_fluxes.f90 File Reference	289
7.2.1 Detailed Description	289
7.2.2 Function/Subroutine Documentation	289
7.2.2.1 add_fluxes()	289
7.3 src/alias.f90 File Reference	290
7.3.1 Detailed Description	290
7.3.2 Function/Subroutine Documentation	290
7.3.2.1 alias()	290
7.4 src/allocate_func3d.f90 File Reference	291
7.4.1 Detailed Description	291
7.4.2 Function/Subroutine Documentation	291
7.4.2.1 allocate_func3d()	292
7.5 src/allocate_ns.f90 File Reference	292
7.5.1 Detailed Description	292
7.5.2 Function/Subroutine Documentation	293
7.5.2.1 allocate_ns()	293
7.6 src/allocate_nunv.f90 File Reference	294
7.6.1 Detailed Description	294
7.6.2 Function/Subroutine Documentation	294
7.6.2.1 allocate_nunv()	294
7.7 src/aspectratio.f90 File Reference	295
7.7.1 Detailed Description	295
7.7.2 Function/Subroutine Documentation	295
7.7.2.1 aspectratio()	295
7.8 src/bcovar.f90 File Reference	296
7.8.1 Detailed Description	296
7.8.2 Function/Subroutine Documentation	296
7.8.2.1 bcovar()	296
7.9 src/bextrema.f90 File Reference	297
7.9.1 Detailed Description	298
7.9.2 Function/Subroutine Documentation	298
7.9.2.1 bextrema()	298
7.10 src/bss.f90 File Reference	298
7.10.1 Detailed Description	299
7.10.2 Function/Subroutine Documentation	299
7.10.2.1 bss()	299

7.11 src/calc_fbal.f90 File Reference	300
7.11.1 Detailed Description	300
7.11.2 Function/Subroutine Documentation	300
7.11.2.1 calc_fbal()	300
7.12 src/CMakeLists.txt File Reference	301
7.12.1 Function Documentation	301
7.12.1.1 add_subdirectory()	301
7.13 src/data/CMakeLists.txt File Reference	301
7.13.1 Function Documentation	302
7.13.1.1 list()	302
7.14 src/NESTOR/CMakeLists.txt File Reference	302
7.14.1 Function Documentation	302
7.14.1.1 add_subdirectory()	302
7.15 src/NESTOR/data/CMakeLists.txt File Reference	302
7.15.1 Function Documentation	302
7.15.1.1 list()	303
7.16 src/convert.f90 File Reference	303
7.16.1 Detailed Description	303
7.16.2 Function/Subroutine Documentation	303
7.16.2.1 convert()	303
7.17 src/data/fbal.f90 File Reference	304
7.18 src/data/realspace.f90 File Reference	304
7.19 src/data/stel_constants.f90 File Reference	305
7.20 src/data/stel_kinds.f90 File Reference	306
7.21 src/data/vforces.f90 File Reference	306
7.22 src/data/vmec_dim.f90 File Reference	307
7.23 src/data/vmec_input.f90 File Reference	307
7.24 src/data/vmec_io.f90 File Reference	308
7.25 src/data/vmec_main.f90 File Reference	309
7.26 src/data/vmec_params.f90 File Reference	312
7.27 src/data/vmec_persistent.f90 File Reference	313
7.28 src/data/vmercier.f90 File Reference	314
7.29 src/data/vparams.f90 File Reference	314
7.30 src/data/vsvd0.f90 File Reference	315
7.31 src/data/xstuff.f90 File Reference	315
7.32 src/elongation.f90 File Reference	316
7.32.1 Detailed Description	316
7.32.2 Function/Subroutine Documentation	316
7.32.2.1 elongation()	316
7.33 src/eqfor.f90 File Reference	317
7.33.1 Detailed Description	317
7.33.2 Function/Subroutine Documentation	317

7.33.2.1 eqfor()	317
7.34 src/eqsolve.f90 File Reference	319
7.34.1 Detailed Description	319
7.34.2 Function/Subroutine Documentation	319
7.34.2.1 eqsolve()	319
7.35 src/evolve.f90 File Reference	320
7.35.1 Detailed Description	321
7.35.2 Function/Subroutine Documentation	321
7.35.2.1 evolve()	321
7.36 src/fileout.f90 File Reference	322
7.36.1 Detailed Description	323
7.36.2 Function/Subroutine Documentation	323
7.36.2.1 fileout()	323
7.37 src/fixaray.f90 File Reference	325
7.37.1 Detailed Description	325
7.37.2 Function/Subroutine Documentation	325
7.37.2.1 fixaray()	325
7.38 src/flip_theta.f90 File Reference	326
7.38.1 Detailed Description	326
7.38.2 Function/Subroutine Documentation	326
7.38.2.1 flip_theta()	326
7.39 src/forces.f90 File Reference	327
7.39.1 Detailed Description	327
7.39.2 Function/Subroutine Documentation	327
7.39.2.1 forces()	327
7.40 src/free_mem_func3d.f90 File Reference	328
7.40.1 Detailed Description	328
7.40.2 Function/Subroutine Documentation	328
7.40.2.1 free_mem_func3d()	328
7.41 src/free_mem_ns.f90 File Reference	329
7.41.1 Detailed Description	329
7.41.2 Function/Subroutine Documentation	329
7.41.2.1 free_mem_ns()	329
7.42 src/free_mem_nunv.f90 File Reference	330
7.42.1 Detailed Description	330
7.42.2 Function/Subroutine Documentation	330
7.42.2.1 free_mem_nunv()	330
7.43 src/freeb_data.f90 File Reference	331
7.43.1 Detailed Description	331
7.43.2 Function/Subroutine Documentation	331
7.43.2.1 freeb_data()	331
7.44 src/fsym_fft.f90 File Reference	332

7.44.1 Detailed Description	332
7.44.2 Function/Subroutine Documentation	332
7.44.2.1 fext_fft()	332
7.44.2.2 fsym_fft()	333
7.45 src/fsym_invfft.f90 File Reference	334
7.45.1 Detailed Description	334
7.45.2 Function/Subroutine Documentation	334
7.45.2.1 fsym_invfft()	334
7.46 src/funct3d.f90 File Reference	335
7.46.1 Detailed Description	335
7.46.2 Function/Subroutine Documentation	335
7.46.2.1 funct3d()	335
7.47 src/functions.f File Reference	337
7.47.1 Detailed Description	337
7.48 src/getbsubs.f90 File Reference	337
7.48.1 Detailed Description	338
7.48.2 Function/Subroutine Documentation	338
7.48.2.1 getbsubs()	338
7.49 src/getcurmid.f90 File Reference	339
7.49.1 Detailed Description	339
7.49.2 Function/Subroutine Documentation	339
7.49.2.1 getcurmid()	339
7.50 src/getfsq.f90 File Reference	340
7.50.1 Detailed Description	340
7.50.2 Function/Subroutine Documentation	340
7.50.2.1 getfsq()	340
7.51 src/guess_axis.f90 File Reference	341
7.51.1 Detailed Description	341
7.51.2 Function/Subroutine Documentation	341
7.51.2.1 guess_axis()	341
7.52 src/heading.f90 File Reference	342
7.52.1 Detailed Description	342
7.52.2 Function/Subroutine Documentation	342
7.52.2.1 heading()	342
7.53 src/initialize_radial.f90 File Reference	343
7.53.1 Detailed Description	343
7.53.2 Function/Subroutine Documentation	343
7.53.2.1 initialize_radial()	343
7.54 src/interp.f90 File Reference	344
7.54.1 Detailed Description	345
7.54.2 Function/Subroutine Documentation	345
7.54.2.1 interp()	345

7.55 src/jacobian.f90 File Reference	346
7.55.1 Detailed Description	346
7.55.2 Function/Subroutine Documentation	346
7.55.2.1 jacobian()	346
7.56 src/jxbforce.f90 File Reference	346
7.56.1 Detailed Description	347
7.56.2 Function/Subroutine Documentation	347
7.56.2.1 jxbforce()	347
7.57 src/lamcal.f90 File Reference	348
7.57.1 Detailed Description	349
7.57.2 Function/Subroutine Documentation	349
7.57.2.1 lamcal()	349
7.58 src/line_segment.f File Reference	349
7.58.1 Detailed Description	350
7.59 src/magnetic_fluxes.f90 File Reference	350
7.59.1 Detailed Description	350
7.59.2 Function/Subroutine Documentation	350
7.59.2.1 polflux()	350
7.59.2.2 polflux_deriv()	351
7.59.2.3 torflux()	352
7.59.2.4 torflux_deriv()	353
7.60 src/mercier.f90 File Reference	354
7.60.1 Detailed Description	354
7.60.2 Function/Subroutine Documentation	354
7.60.2.1 mercier()	354
7.61 src/mgrid_mod.f File Reference	355
7.61.1 Detailed Description	357
7.62 src/NESTOR/analysum.f90 File Reference	358
7.62.1 Function/Subroutine Documentation	358
7.62.1.1 analysum()	358
7.63 src/NESTOR/analysum2.f90 File Reference	358
7.63.1 Function/Subroutine Documentation	359
7.63.1.1 analysum2()	359
7.64 src/NESTOR/analyt.f90 File Reference	359
7.64.1 Function/Subroutine Documentation	359
7.64.1.1 analyt()	360
7.65 src/NESTOR/becoil.f90 File Reference	360
7.65.1 Function/Subroutine Documentation	361
7.65.1.1 becoil()	361
7.66 src/NESTOR/belicu.f90 File Reference	361
7.66.1 Function/Subroutine Documentation	361
7.66.1.1 belicu()	362

7.67 src/NESTOR/bextern.f90 File Reference	362
7.67.1 Function/Subroutine Documentation	362
7.67.1.1 bextern()	362
7.68 src/NESTOR/data/nestor_io.f90 File Reference	363
7.68.1 Detailed Description	366
7.68.2 Function/Subroutine Documentation	366
7.68.2.1 read_nestor_outputs()	367
7.68.2.2 write_nestor_inputs()	367
7.69 src/NESTOR/data/vac_persistent.f90 File Reference	368
7.70 src/NESTOR/data/vacmod.f90 File Reference	369
7.71 src/NESTOR/data/vacmod0.f90 File Reference	371
7.72 src/NESTOR/fouri.f90 File Reference	372
7.72.1 Function/Subroutine Documentation	372
7.72.1.1 fouri()	372
7.73 src/NESTOR/fourp.f90 File Reference	373
7.73.1 Function/Subroutine Documentation	373
7.73.1.1 fourp()	373
7.74 src/NESTOR/greenf.f90 File Reference	373
7.74.1 Function/Subroutine Documentation	373
7.74.1.1 greenf()	374
7.75 src/NESTOR/nestor_main.f90 File Reference	374
7.75.1 Detailed Description	374
7.75.2 Function/Subroutine Documentation	374
7.75.2.1 nestor()	375
7.76 src/NESTOR/precalf.f90 File Reference	375
7.76.1 Function/Subroutine Documentation	375
7.76.1.1 precalf()	376
7.77 src/NESTOR/scalpot.f90 File Reference	376
7.77.1 Function/Subroutine Documentation	376
7.77.1.1 scalpot()	376
7.78 src/NESTOR/surface.f90 File Reference	377
7.78.1 Function/Subroutine Documentation	377
7.78.1.1 surface()	378
7.79 src/NESTOR/vacuum.f90 File Reference	378
7.79.1 Function/Subroutine Documentation	378
7.79.1.1 vacuum()	379
7.80 src/open_output_files.f90 File Reference	380
7.80.1 Detailed Description	380
7.80.2 Function/Subroutine Documentation	380
7.80.2.1 open_output_files()	380
7.81 src/parse_extension.f File Reference	381
7.81.1 Detailed Description	381

7.81.2 Function/Subroutine Documentation	381
7.81.2.1 parse_extension()	381
7.82 src/precondn.f90 File Reference	382
7.82.1 Detailed Description	382
7.82.2 Function/Subroutine Documentation	382
7.82.2.1 precondn()	383
7.83 src/printout.f90 File Reference	384
7.83.1 Detailed Description	384
7.83.2 Function/Subroutine Documentation	384
7.83.2.1 printout()	384
7.84 src/profil1d.f90 File Reference	385
7.84.1 Detailed Description	385
7.84.2 Function/Subroutine Documentation	385
7.84.2.1 profil1d()	386
7.85 src/profil3d.f90 File Reference	386
7.85.1 Detailed Description	386
7.85.2 Function/Subroutine Documentation	387
7.85.2.1 profil3d()	387
7.86 src/profile_functions.f File Reference	387
7.86.1 Detailed Description	388
7.86.2 Function/Subroutine Documentation	388
7.86.2.1 pcurr()	388
7.86.2.2 piota()	389
7.86.2.3 pmass()	390
7.87 src/read_indata.f90 File Reference	390
7.87.1 Detailed Description	390
7.87.2 Function/Subroutine Documentation	390
7.87.2.1 read_indata()	390
7.88 src/read_wout_mod.f File Reference	391
7.88.1 Detailed Description	399
7.89 src/readin.f90 File Reference	399
7.89.1 Detailed Description	400
7.89.2 Function/Subroutine Documentation	400
7.89.2.1 readin()	400
7.90 src/reset_params.f90 File Reference	401
7.90.1 Function/Subroutine Documentation	401
7.90.1.1 reset_params()	402
7.91 src/residue.f90 File Reference	402
7.91.1 Function/Subroutine Documentation	402
7.91.1.1 constrain_m1()	403
7.91.1.2 residue()	403
7.91.1.3 scale_m1()	404

7.92 src/restart_iter.f90 File Reference	405
7.92.1 Function/Subroutine Documentation	405
7.92.1.1 restart_iter()	405
7.93 src/safe_open_mod.f File Reference	405
7.94 src/scalfor.f90 File Reference	406
7.94.1 Function/Subroutine Documentation	406
7.94.1.1 scalfor()	406
7.95 src/solver.f90 File Reference	407
7.95.1 Function/Subroutine Documentation	407
7.95.1.1 solver()	407
7.96 src/spectrum.f90 File Reference	407
7.96.1 Function/Subroutine Documentation	407
7.96.1.1 spectrum()	408
7.97 src/spline_akima.f File Reference	408
7.97.1 Function/Subroutine Documentation	408
7.97.1.1 spline_akima()	409
7.98 src/spline_akima_int.f File Reference	409
7.98.1 Function/Subroutine Documentation	409
7.98.1.1 spline_akima_int()	409
7.99 src/spline_cubic.f File Reference	410
7.99.1 Function/Subroutine Documentation	410
7.99.1.1 spline_cubic()	410
7.99.1.2 spline_nr()	411
7.99.1.3 splint_nr()	412
7.100 src/spline_cubic_int.f File Reference	412
7.100.1 Function/Subroutine Documentation	412
7.100.1.1 spline_cubic_int()	412
7.100.1.2 spline_int()	413
7.100.1.3 splint_int()	414
7.101 src/symforce.f90 File Reference	414
7.101.1 Function/Subroutine Documentation	414
7.101.1.1 symforce()	415
7.101.1.2 symoutput()	415
7.102 src/symrzl.f90 File Reference	416
7.102.1 Function/Subroutine Documentation	416
7.102.1.1 symrzl()	416
7.103 src/tolower.f90 File Reference	417
7.103.1 Function/Subroutine Documentation	417
7.103.1.1 tolower()	417
7.104 src/tomnsp.f90 File Reference	418
7.104.1 Function/Subroutine Documentation	418
7.104.1.1 tomnspa()	418

7.104.1.2 tomnsps()	419
7.105 src/totzsp.f90 File Reference	419
7.105.1 Function/Subroutine Documentation	420
7.105.1.1 convert_asym()	420
7.105.1.2 convert_sym()	420
7.105.1.3 totzspa()	421
7.105.1.4 totzsps()	422
7.106 src/tridslv.f90 File Reference	423
7.106.1 Function/Subroutine Documentation	423
7.106.1.1 tridslv()	423
7.107 src/vmec.f90 File Reference	424
7.107.1 Detailed Description	424
7.107.2 Function/Subroutine Documentation	424
7.107.2.1 vmec()	424
7.107.2.2 xvmec()	425
7.108 src/wrout.f90 File Reference	426
7.108.1 Function/Subroutine Documentation	426
7.108.1.1 wrout()	426
Index	429

Chapter 1

Educational VMEC

This is a heavily stripped-down version of the serial implementation of VMEC 8.52. It is forked from the `v251` branch of the `STELLOPT` repository.

The goal of this project is to have a version of VMEC which only computes the Stellarator MHD equilibrium and nothing more.

The `cmake` build system for stand-alone VMEC is borrowed from `hiddenSymmetries/VMEC2000` and from `ORNL-Fusion/LIBSTELL`.

1.1 Building

This is a fairly standard CMake setup, if you are used to it. Here is how it works:

- Create a directory `build` in the main folder: `mkdir build`
- Go into the `build` directory: `cd build`
- Run CMake: `cmake ..`
- Execute the actual build process: `make` (optional multi-threaded build: `make -j`)
- The VMEC executable `xvmec` is then located in `build/bin` with respect to the main folder.

1.2 Example Execution

- Change into the `test` dir: `cd test`
- Run the `Solov'ev` test case: `../build/bin/xvmec input.solovev`

1.3 External NESTOR

The free-boundary part of VMEC is the Neumann Solver for Toroidal Systems (NESTOR). Its source code is in a separate folder `NESTOR`. The appropriate reference is [https://doi.org/10.1016/0021-9991\(86\)90055-0](https://doi.org/10.1016/0021-9991(86)90055-0).

This version of NESTOR can be run stand-alone. It reads its inputs from a netCDF file and writes its outputs into another netCDF file. The main executable of this stand-alone version of NESTOR is `nestor_main.f90`. The input and output files are read and written in `nestor_io.f90`.

This version of VMEC can be configured to dump the corresponding input and output files, but still run the compiled-in version of NESTOR. This is enabled via the logical flag `ldump_vacuum_ref` in `funct3d.f90`.

Also, an external NESTOR implementation can be called instead of using the compiled-in version of NESTOR. This is enabled via the logical flag `lexternal_nestor` in `funct3d.f90`. The corresponding system call to execute the external NESTOR implementation has to be specified in `nestor_executable` in `funct3d.f90`.

1.4 Angle Constraint

The poloidal angle-like coordinate is a priori not uniquely defined and needs special care. The version of VMEC from the STELLOPT repo had essentially two options for this. They were alternatively compiled in via the preprocessor flag `_HBANGLE`.

1. The Hirshman-Breslau explicit spectrally optimized Fourier series (see <https://doi.org/10.1063/1.872954> for details) and
2. an unknown mixture of several constraints of the $m=1$ Fourier coefficients (the logical `lconm1` is true for this constraint).

By default, the `_HBANGLE` preprocessor flag is not active and thus, the "old" $m=1$ constraint is active.

This version of VMEC has most, if not all, of its preprocessor flags explicitly expanded. It became clear that it is nevertheless useful to have at least a vague idea of what parts of the code are related to the angle constraint. Therefore, those parts of VMEC related to the $m=1$ constraint are marked to start with

```
! #ifndef _hbangle
```

and end with

```
! #end /* ndef _HBANGLE */
```

Chapter 2

Modules Index

2.1 Modules List

Here is a list of all modules with brief descriptions:

fbal	11
functions	12
line_segment	
This module contains code to create a profile constructed of line segments. These line segments are assumed to be specified such that $xx(i) < xx(i + 1)$	15
mgrid_mod	
Precomputed table of magnetic field due to confinement coils	17
nestor_io	
Input and Output for stand-alone NESTOR	37
read_wout_mod	
Reading of <code>wout</code> VMEC output file	71
realspace	161
safe_open_mod	167
stel_constants	168
stel_kinds	169
vac_persistent	171
vacmod	177
vacmod0	202
vforces	207
vmec_dim	212
vmec_input	215
vmec_io	228
vmec_main	232
vmec_params	260
vmec_persistent	270
vmecier	275
vparams	276
vsvd0	281
xstuff	282

Chapter 3

Data Type Index

3.1 Data Types List

Here are the data types with brief descriptions:

[read_wout_mod::read_wout_file](#) 285

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

src/add_fluxes.f90	Add the magnetic fluxes to the tangential derivatives of λ to arrive at the contravariant magnetic field components B^θ and B^ζ	289
src/alias.f90	Fourier transform alias force and also return intermediate output	290
src/allocate_funct3d.f90	Allocate arrays required in <code>funct3d()</code>	291
src/allocate_ns.f90	Allocate arrays depending on the number of flux surfaces <code>ns</code>	292
src/allocate_nunv.f90	Allocate arrays depending on the number of Fourier coefficients <code>nunv</code>	294
src/aspectratio.f90	Compute aspect-ratio (independent of elongation): $A = \langle R \rangle / \sqrt{\langle ab \rangle}$	295
src/bcovar.f90	Compute the covariant components of the magnetic field B_θ, B_ζ	296
src/bextrema.f90	Computes minimum and maximum $ \mathbf{B} $ along ζ between two angle lines ($\theta = 0, \pi$)	297
src/bss.f90	Computes <code>br</code> , <code>bphi</code> , <code>bz</code> , <code>bsubs</code> on half-radial mesh	298
src/calc_fbal.f90	Compute flux-surface averaged radial force balance $\nabla p - \langle \mathbf{j} \times \mathbf{B} \rangle$	300
src/convert.f90	Convert internal mode representation to standard form for output (coefficients of $\cos(\mu-nv)$, $\sin(\mu-nv)$ without internal <code>mscale</code> , <code>nscale</code> norms)	303
src/elongation.f90	Compute Waist thickness and height in $\varphi = 0, \pi$ symmetry planes	316
src/eqfor.f90	Basis physics analysis and evaluation of force balance. This is where most of the contents of the <code>threed1</code> output file is computed	317
src/eqsolve.f90	Iteratively evolve the Fourier coefficients that specify the equilibrium	319
src/evolve.f90	Take a single time step in Fourier space to evolve the Fourier coefficients describing the equilibrium towards force balance	320
src/fileout.f90	Write the output files	322

src/fixaray.f90	Allocate and fill some fixed-size arrays (only depending on Fourier resolution)	325
src/flip_theta.f90	Flip the definition of the poloidal angle in the user-provided initial guess for the LCFS geometry	326
src/forces.f90	Compute the real-space MHD forces	327
src/free_mem_funct3d.f90	Free memory required by <code>funct3d()</code>	328
src/free_mem_ns.f90	Free memory depending on the number of flux surfaces <code>ns</code>	329
src/free_mem_nunv.f90	Free arrays depending on the number of Fourier coefficients <code>nunv</code>	330
src/freeb_data.f90	Write out edge values of fields	331
src/fsym_fft.f90	Fourier transforms	332
src/fsym_invfft.f90	Extends function from <code>ntheta2</code> to <code>ntheta3</code> range	334
src/funct3d.f90	Evaluate the three-dimensional MHD energy functional	335
src/functions.f	This module contains functions used by the profiles	337
src/getbsubs.f90	Solves the radial force balance $\mathbf{B} \cdot \mathbf{B}_s = F_s$ for B_s in real space using collocation	337
src/getcurmid.f90	Get current at midplane (?)	339
src/getfsq.f90	Compute total force residual on flux surfaces	340
src/guess_axis.f90	Computes guess for magnetic axis if user guess leads to initial sign change of Jacobian	341
src/heading.f90	Open output files and print banner message at the top	342
src/initialize_radial.f90	Allocates memory for radial arrays and initializes radial profiles	343
src/interp.f90	Interpolate R , Z and λ on full grid	344
src/jacobian.f90	Evaluate the Jacobian of the transform from flux- to cylindrical coordinates	346
src/jxbforce.f90	Program for computing local $\mathbf{K} \times \mathbf{B} = \nabla p$ force balance	346
src/lamcal.f90	Normalization parameters for λ	348
src/line_segment.f	This module contains code to create a profile constructed of line segments	349
src/magnetic_fluxes.f90	Compute toroidal and poloidal magnetic flux profiles	350
src/mercier.f90	Evaluate the Mercier stability criterion	354
src/mgrid_mod.f	Precomputed table of magnetic field due to confinement coils	355
src/open_output_files.f90	Open output files	380
src/parse_extension.f	Parse the first command-line argument into a filename	381
src/precondn.f90	Compute preconditioning matrix elements for R , Z force	382
src/printout.f90	Print iteration progress to screen and <code>threed1</code> output file	384

src/profil1d.f90	
Compute phip and iota profiles on full grid	385
src/profil3d.f90	
Compute three-dimensional profiles of flux-surface geometry etc	386
src/profile_functions.f	
Parameterizations of profiles for rotational transform, toroidal current and pressure/mass	387
src/read_indata.f90	
Read the INDATA namelist from a given input file	390
src/read_wout_mod.f	
Reading of wout VMEC output file	391
src/readin.f90	
Read the input file	399
src/reset_params.f90	401
src/residue.f90	402
src/restart_iter.f90	405
src/safe_open_mod.f	405
src/scalfor.f90	406
src/solver.f90	407
src/spectrum.f90	407
src/spline_akima.f	408
src/spline_akima_int.f	409
src/spline_cubic.f	410
src/spline_cubic_int.f	412
src/symforce.f90	414
src/symrzi.f90	416
src/tolower.f90	417
src/tomnsp.f90	418
src/totzsp.f90	419
src/tridslv.f90	423
src/vmec.f90	
Main program of VMEC	424
src/wrout.f90	426
src/data/fbal.f90	304
src/data/realspace.f90	304
src/data/stel_constants.f90	305
src/data/stel_kinds.f90	306
src/data/vforces.f90	306
src/data/vmec_dim.f90	307
src/data/vmec_input.f90	307
src/data/vmec_io.f90	308
src/data/vmec_main.f90	309
src/data/vmec_params.f90	312
src/data/vmec_persistent.f90	313
src/data/vmercier.f90	314
src/data/vparams.f90	314
src/data/vsvd0.f90	315
src/data/xstuff.f90	315
src/NESTOR/analysum.f90	358
src/NESTOR/analysum2.f90	358
src/NESTOR/analyt.f90	359
src/NESTOR/becoil.f90	360
src/NESTOR/belicu.f90	361
src/NESTOR/bextern.f90	362
src/NESTOR/fouri.f90	372
src/NESTOR/fourp.f90	373
src/NESTOR/greenf.f90	373
src/NESTOR/nestor_main.f90	
Main program of stand-alone version of NESTOR	374

src/NESTOR/ precal.f90	375
src/NESTOR/ scalpot.f90	376
src/NESTOR/ surface.f90	377
src/NESTOR/ vacuum.f90	378
src/NESTOR/data/ nestor_io.f90	
Input and Output for stand-alone NESTOR	363
src/NESTOR/data/ vac_persistent.f90	368
src/NESTOR/data/ vacmod.f90	369
src/NESTOR/data/ vacmod0.f90	371

Chapter 5

Module Documentation

5.1 fbal Module Reference

Variables

- `real(dp), dimension(:), allocatable rzu_fac`
- `real(dp), dimension(:), allocatable rru_fac`
- `real(dp), dimension(:), allocatable frcc_fac`
- `real(dp), dimension(:), allocatable fzsc_fac`

5.1.1 Variable Documentation

5.1.1.1 frcc_fac

`real(dp), dimension(:), allocatable fbal::frcc_fac`

Definition at line 10 of file fbal.f90.

Referenced by `allocate_ns()`, `bcovar()`, and `free_mem_ns()`.

5.1.1.2 fzsc_fac

`real(dp), dimension(:), allocatable fbal::fzsc_fac`

Definition at line 11 of file fbal.f90.

Referenced by `allocate_ns()`, `bcovar()`, and `free_mem_ns()`.

5.1.1.3 rru_fac

```
real(dp), dimension(:), allocatable fbal::rru_fac
```

Definition at line 9 of file fbal.f90.

Referenced by `allocate_ns()`, `bcovar()`, and `free_mem_ns()`.

5.1.1.4 rzu_fac

```
real(dp), dimension(:), allocatable fbal::rzu_fac
```

Definition at line 8 of file fbal.f90.

Referenced by `allocate_ns()`, `bcovar()`, and `free_mem_ns()`.

5.2 functions Module Reference

Functions/Subroutines

- `real(rprec)` function, public `two_power` (x, b)
*Profile function for the two_power profile. $b(0) * (1 - x^{b(1)})^{b(2)}$.*
- `real(rprec)` function, public `two_power_gs` (x, b)
*Profile function for the two_power_gs profile. $two_power(x) * (1 + \sum [b(i) * \exp(-(x - b(i + 1))/b(i + 2))^2])$.*
- logical function `function_test` ()
Main test function.

5.2.1 Function/Subroutine Documentation

5.2.1.1 function_test()

```
logical function functions::function_test
```

Main test function.

Test `two_power` function for x = 0, b = {1,10,2} is 1

Test `two_power` function for x = 1, b = {1,10,2} is 0

Test `two_power` function for x = 0.5, b = {1,1,1} is 0.5

Test `two_power` function for x = 0.5, b = {1,1,2} is 0.25

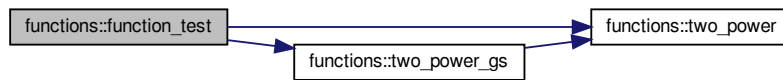
Test `two_power_gs` function for x = 0.4, b = {1,1,1,0,0,1} is `two_power(x,b)`

Test `two_power_gs` function for $x = 0.8$, $b = \{1,1,0,1,0.8,0.1\}$ is 2

Definition at line 51 of file `functions.f`.

References `two_power()`, and `two_power_gs()`.

Here is the call graph for this function:



5.2.1.2 `two_power()`

```

real(rprec) function, public functions::two_power (
    real(rprec), intent(in) x,
    real(rprec), dimension(0:20), intent(in) b )
  
```

Profile function for the `two_power` profile. $b(0) * (1 - x^{b(1)})^{b(2)}$.

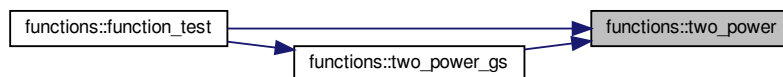
Parameters

x	evaluation location
b	parameter vector

Definition at line 20 of file `functions.f`.

Referenced by `function_test()`, and `two_power_gs()`.

Here is the caller graph for this function:



5.2.1.3 two_power_gs()

```
real(rprec) function, public functions::two_power_gs (  
    real(rprec), intent(in) x,  
    real(rprec), dimension(0:20), intent(in) b )
```

Profile function for the two_power_gs profile. $\text{two_power}(x) * (1 + \sum [b(i) * \exp(-(x - b(i + 1))/b(i + 2))^2])$.

Parameters

x	evaluation location
b	parameter vector

Definition at line 34 of file functions.f.

References two_power().

Referenced by function_test().

Here is the call graph for this function:



Here is the caller graph for this function:



5.3 line_segment Module Reference

This module contains code to create a profile constructed of line segments. These line segments are assumed to be specified such that $xx(i) < xx(i + 1)$.

Functions/Subroutines

- subroutine, public [line_seg](#) (x, y, xx, yy, n)
- subroutine, public [line_seg_int](#) (x, y, xx, yy, n)
- logical function, public [line_seg_test](#) ()

5.3.1 Detailed Description

This module contains code to create a profile constructed of line segments. These line segments are assumed to be specified such that $xx(i) < xx(i + 1)$.

5.3.2 Function/Subroutine Documentation

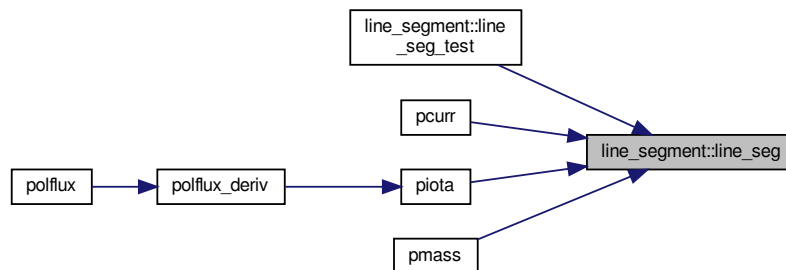
5.3.2.1 line_seg()

```
subroutine, public line_segment::line_seg (
    real(rprec), intent(in) x,
    real(rprec), intent(out) y,
    real(rprec), dimension(n), intent(in) xx,
    real(rprec), dimension(n), intent(in) yy,
    integer, intent(in) n )
```

Definition at line 31 of file line_segment.f.

Referenced by line_seg_test(), pcurr(), piota(), and pmass().

Here is the caller graph for this function:



5.3.2.2 line_seg_int()

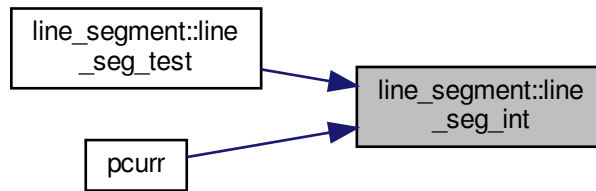
```
subroutine, public line_segment::line_seg_int (
    real(rprec), intent(in) x,
    real(rprec), intent(out) y,
    real(rprec), dimension(n), intent(in) xx,
    real(rprec), dimension(n), intent(in) yy,
    integer, intent(in) n )
```

Definition at line 74 of file line_segment.f.

References stel_constants::zero.

Referenced by line_seg_test(), and pcurr().

Here is the caller graph for this function:



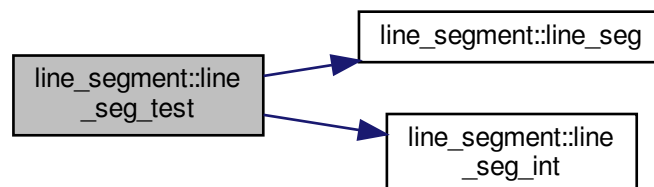
5.3.2.3 line_seg_test()

logical function, public line_segment::line_seg_test

Definition at line 255 of file line_segment.f.

References `line_seg()`, and `line_seg_int()`.

Here is the call graph for this function:



5.4 mgrid_mod Module Reference

Precomputed table of magnetic field due to confinement coils.

Functions/Subroutines

- subroutine [read_mgrid](#) (mgrid_file, extcur, nv, nfp, lscreen, ier_flag)
- subroutine [sum_bfield](#) (bfield, bf_add, cur, n1)
- subroutine [assign_bptrs](#) (bptr)
- subroutine [free_mgrid](#) (istat)

Variables

- integer, parameter `nlimset` = 2
- character(len= *), parameter `vn_br0` = 'br'
- character(len= *), parameter `vn_bp0` = 'bp'
- character(len= *), parameter `vn_bz0` = 'bz'
- character(len= *), parameter `vn_ir` = 'ir'
- character(len= *), parameter `vn_jz` = 'jz'
- character(len= *), parameter `vn_kp` = 'kp'
- character(len= *), parameter `vn_nfp` = 'nfp'
- character(len= *), parameter `vn_rmin` = 'rmin'
- character(len= *), parameter `vn_rmax` = 'rmax'
- character(len= *), parameter `vn_zmin` = 'zmin'
- character(len= *), parameter `vn_zmax` = 'zmax'
- character(len= *), parameter `vn_coilgrp` = 'coil_group'
- character(len= *), parameter `vn_nextcur` = 'nextcur'
- character(len= *), parameter `vn_mgmode` = 'mgrid_mode'
- character(len= *), parameter `vn_coilcur` = 'raw_coil_cur'
- character(len= *), parameter `ln_next` = 'External currents'
- integer `nr0b`
- integer `np0b`
- integer `nfper0`
- integer `nz0b`
- integer `nobd`
- integer `nobser`
- integer `nextcur`
- integer `nbfldn`
- integer `nbsets`
- integer `nbcoilsn`
- integer `nbvac`
- integer `nbcoil_max`
- integer `nlim`
- integer `nlim_max`
- integer `nsets`
- integer `nrgrid`
- integer `nzgrid`
- integer, dimension(:), allocatable `needflx`
- integer, dimension(:), allocatable `nbcoils`
- integer, dimension(:), allocatable `limitr`
- integer, dimension(:), allocatable `nsetsn`
- integer, dimension(:, :), allocatable `iconnect`
- integer, dimension(:, :), allocatable `needbfld`
- real(rprec) `rminb`
- real(rprec) `zminb`
- real(rprec) `rmaxb`
- real(rprec) `zmaxb`
- real(rprec) `delrb`
- real(rprec) `delzb`
- real(rprec) `rx1`
- real(rprec) `rx2`
- real(rprec) `zy1`
- real(rprec) `zy2`
- real(rprec) `condif`
- real(rprec), dimension(:, :), allocatable, target `bvac`
- real(rprec), dimension(:, :, :), pointer `brvac`

- real(rprec), dimension(:,:), pointer [bzvac](#)
- real(rprec), dimension(:,:), pointer [bpvac](#)
- real(rprec), dimension(:,:), allocatable [unpsiext](#)
- real(rprec), dimension(:,:), allocatable [plbfld](#)
- real(rprec), dimension(:,:), allocatable [rbcoil](#)
- real(rprec), dimension(:,:), allocatable [zbcoil](#)
- real(rprec), dimension(:,:), allocatable [abcoil](#)
- real(rprec), dimension(:,:), allocatable [bcoil](#)
- real(rprec), dimension(:,:), allocatable [rbcoilsqr](#)
- real(rprec), dimension(:), allocatable [raw_coil_current](#)
- real(rprec), dimension(:), allocatable [xobser](#)
- real(rprec), dimension(:), allocatable [zobser](#)
- real(rprec), dimension(:), allocatable [xobsqr](#)
- real(rprec), dimension(:), allocatable [dsiext](#)
- real(rprec), dimension(:), allocatable [psiext](#)
- real(rprec), dimension(:), allocatable [plflux](#)
- real(rprec), dimension(:), allocatable [b_chi](#)
- character(len=300) [mgrid_path](#)
- character(len=300) [mgrid_path_old](#) = " "
- character(len=30), dimension(:), allocatable [curlabel](#)
- character(len=15), dimension(:), allocatable [dsilabel](#)
- character(len=15), dimension(:), allocatable [bloopnames](#)
- character(len=30) [tokid](#)
- real(rprec), dimension(:,:), allocatable [dbcoil](#)
- real(rprec), dimension(:,:), allocatable [pfcspec](#)
- real(rprec), dimension(:,:), allocatable [rlim](#)
- real(rprec), dimension(:,:), allocatable [zlim](#)
- real(rprec), dimension(:,:), allocatable [reslim](#)
- real(rprec), dimension(:,:), allocatable [seplim](#)
- character(len=1) [mgrid_mode](#)

5.4.1 Detailed Description

Precomputed table of magnetic field due to confinent coils.

5.4.2 Function/Subroutine Documentation

5.4.2.1 `assign_bptrs()`

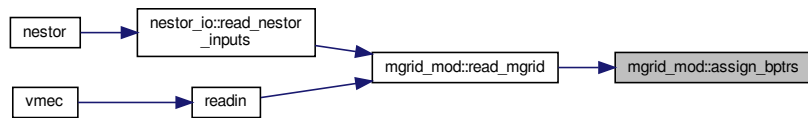
```
subroutine mgrid_mod::assign_bptrs (
    real(rprec), dimension(nr0b,nz0b,np0b,3), intent(in), target bptr )
```

Definition at line 369 of file `mgrid_mod.f`.

References [bpvac](#), [brvac](#), and [bzvac](#).

Referenced by `read_mgrid()`.

Here is the caller graph for this function:



5.4.2.2 free_mgrid()

```

subroutine mgrid_mod::free_mgrid (
    integer istat )

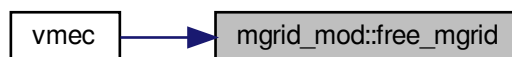
```

Definition at line 379 of file `mgrid_mod.f`.

References `abcoil`, `b_chi`, `bcoil`, `bloopnames`, `bvac`, `curlabel`, `dbcoil`, `dsiext`, `dsilabel`, `iconnect`, `mgrid_path_old`, `nbcoils`, `needbfld`, `needflx`, `pfcspec`, `plbfld`, `plflux`, `psiext`, `raw_coil_current`, `rbcoil`, `rbcoilsqr`, `reslim`, `rlim`, `seplim`, `unpsiext`, `xobser`, `xobsqr`, `zbcoid`, `zlim`, and `zobser`.

Referenced by `vmec()`.

Here is the caller graph for this function:



5.4.2.3 read_mgrid()

```

subroutine mgrid_mod::read_mgrid (
    character(len=*), intent(in) mgrid_file,
    real(rprec), dimension(:), intent(in) extcur,
    integer, intent(in) nv,
    integer, intent(in) nfp,
    logical, intent(in) lscreen,
    integer, intent(out) ier_flag )

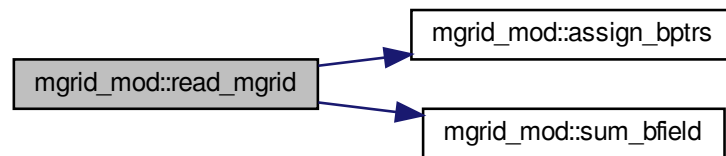
```

Definition at line 105 of file `mgrid_mod.f`.

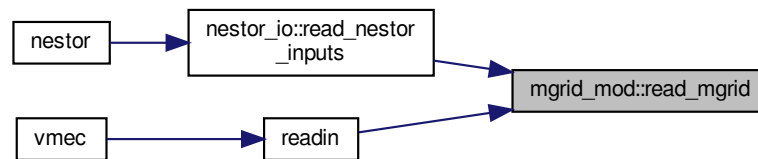
References `assign_bptrs()`, `bvac`, `curlabel`, `delrb`, `delzb`, `vmec_input::lfreeb`, `mgrid_mode`, `mgrid_path`, `mgrid_path_old`, `nbsets`, `nbvac`, `nextcur`, `nfper0`, `vsvd0::nigroup`, `nobd`, `nobser`, `np0b`, `nr0b`, `nz0b`, `raw_coil_current`, `rmaxb`, `rminb`, `sum_bfield()`, `vn_bp0`, `vn_br0`, `vn_bz0`, `vn_coilcur`, `vn_coilgrp`, `vn_ir`, `vn_jz`, `vn_kp`, `vn_mgmode`, `vn_nextcur`, `vn_nfp`, `vn_rmax`, `vn_rmin`, `vn_zmax`, `vn_zmin`, `zmaxb`, and `zminb`.

Referenced by `nestor_io::read_nestor_inputs()`, and `readin()`.

Here is the call graph for this function:



Here is the caller graph for this function:



5.4.2.4 sum_bfield()

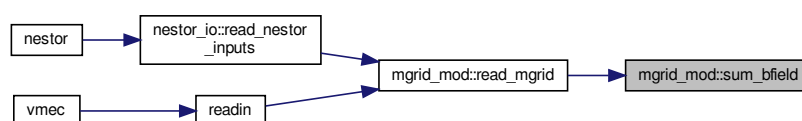
```

subroutine mgrid_mod::sum_bfield (
    real(rprec), dimension(nl), intent(inout) bfield,
    real(rprec), dimension(nl), intent(in) bf_add,
    real(rprec) cur,
    integer nl )
  
```

Definition at line 359 of file `mgrid_mod.f`.

Referenced by `read_mgrid()`.

Here is the caller graph for this function:



5.4.3 Variable Documentation

5.4.3.1 abcoil

```
real(rprec), dimension(:,:), allocatable mgrid_mod::abcoil
```

Definition at line 85 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.2 b_chi

```
real(rprec), dimension(:), allocatable mgrid_mod::b_chi
```

Definition at line 88 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.3 bcoil

```
real(rprec), dimension(:,:), allocatable mgrid_mod::bcoil
```

Definition at line 85 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.4 bloopnames

```
character(len=15), dimension(:), allocatable mgrid_mod::bloopnames
```

Definition at line 93 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.5 bpvac

```
real(rprec), dimension(:,:,:), pointer mgrid_mod::bpvac
```

Definition at line 84 of file mgrid_mod.f.

Referenced by assign_bptrs().

5.4.3.6 brvac

```
real(rprec), dimension(:,:), pointer mgrid_mod::brvac
```

Definition at line 84 of file mgrid_mod.f.

Referenced by assign_bptrs().

5.4.3.7 bvac

```
real(rprec), dimension(:,:), allocatable, target mgrid_mod::bvac
```

Definition at line 83 of file mgrid_mod.f.

Referenced by bextern(), free_mgrid(), and read_mgrid().

5.4.3.8 bzvac

```
real(rprec), dimension(:,:), pointer mgrid_mod::bzvac
```

Definition at line 84 of file mgrid_mod.f.

Referenced by assign_bptrs().

5.4.3.9 condif

```
real(rprec) mgrid_mod::condif
```

Definition at line 82 of file mgrid_mod.f.

5.4.3.10 curlabel

```
character(len=30), dimension(:), allocatable mgrid_mod::curlabel
```

Definition at line 92 of file mgrid_mod.f.

Referenced by free_mgrid(), read_mgrid(), read_wout_mod::read_wout_deallocate(), readin(), read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.4.3.11 dbcoil

```
real(rprec), dimension(:,:), allocatable mgrid_mod::dbcoil
```

Definition at line 96 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.12 delrb

```
real(rprec) mgrid_mod::delrb
```

Definition at line 81 of file mgrid_mod.f.

Referenced by becoil(), and read_mgrid().

5.4.3.13 delzb

```
real(rprec) mgrid_mod::delzb
```

Definition at line 81 of file mgrid_mod.f.

Referenced by becoil(), and read_mgrid().

5.4.3.14 dsiext

```
real(rprec), dimension(:), allocatable mgrid_mod::dsiext
```

Definition at line 88 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.15 dsilabel

```
character(len=15), dimension(:), allocatable mgrid_mod::dsilabel
```

Definition at line 93 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.16 iconnect

```
integer, dimension(:,:), allocatable mgrid_mod::iconnect
```

Definition at line 80 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.17 limitr

```
integer, dimension(:), allocatable mgrid_mod::limitr
```

Definition at line 79 of file mgrid_mod.f.

5.4.3.18 ln_next

```
character(len=*), parameter mgrid_mod::ln_next = 'External currents'
```

Definition at line 17 of file mgrid_mod.f.

5.4.3.19 mgrid_mode

```
character(len=1) mgrid_mod::mgrid_mode
```

Definition at line 99 of file mgrid_mod.f.

Referenced by read_mgrid(), read_wout_mod::read_wout_file::readw_and_open(), reset_params(), and wrout().

5.4.3.20 mgrid_path

```
character(len=300) mgrid_mod::mgrid_path
```

Definition at line 90 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.21 mgrid_path_old

```
character(len=300) mgrid_mod::mgrid_path_old = " "
```

Definition at line 91 of file mgrid_mod.f.

Referenced by free_mgrid(), and read_mgrid().

5.4.3.22 nbcoil_max

```
integer mgrid_mod::nbcoil_max
```

Definition at line 76 of file mgrid_mod.f.

5.4.3.23 nbcoils

```
integer, dimension(:), allocatable mgrid_mod::nbcoils
```

Definition at line 78 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.24 nbcoilsn

```
integer mgrid_mod::nbcoilsn
```

Definition at line 75 of file mgrid_mod.f.

5.4.3.25 nbfldn

```
integer mgrid_mod::nbfldn
```

Definition at line 75 of file mgrid_mod.f.

5.4.3.26 nbsets

```
integer mgrid_mod::nbsets
```

Definition at line 75 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.27 nbvac

```
integer mgrid_mod::nbvac
```

Definition at line 76 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.28 needbfld

```
integer, dimension(:,:), allocatable mgrid_mod::needbfld
```

Definition at line 80 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.29 needflx

```
integer, dimension(:), allocatable mgrid_mod::needflx
```

Definition at line 78 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.30 nextcur

```
integer mgrid_mod::nextcur
```

Definition at line 75 of file mgrid_mod.f.

Referenced by read_mgrid(), readin(), read_wout_mod::read_wout_file::readw_and_open(), reset_params(), write_nestor_inputs(), and wrout().

5.4.3.31 nfper0

```
integer mgrid_mod::nfper0
```

Definition at line 74 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.32 nlim

```
integer mgrid_mod::nlim
```

Definition at line 76 of file mgrid_mod.f.

5.4.3.33 nlim_max

```
integer mgrid_mod::nlim_max
```

Definition at line 76 of file mgrid_mod.f.

5.4.3.34 nlimset

```
integer, parameter mgrid_mod::nlimset = 2
```

Definition at line 10 of file mgrid_mod.f.

5.4.3.35 nobd

```
integer mgrid_mod::nobd
```

Definition at line 75 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.36 nobser

```
integer mgrid_mod::nobser
```

Definition at line 75 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.37 np0b

```
integer mgrid_mod::np0b
```

Definition at line 74 of file mgrid_mod.f.

Referenced by becoil(), read_mgrid(), and readin().

5.4.3.38 nr0b

```
integer mgrid_mod::nr0b
```

Definition at line 74 of file mgrid_mod.f.

Referenced by becoil(), read_mgrid(), and readin().

5.4.3.39 nrgrid

```
integer mgrid_mod::nrgrid
```

Definition at line 76 of file mgrid_mod.f.

5.4.3.40 nsets

```
integer mgrid_mod::nsets
```

Definition at line 76 of file mgrid_mod.f.

5.4.3.41 nsetsn

```
integer, dimension(:), allocatable mgrid_mod::nsetsn
```

Definition at line 79 of file mgrid_mod.f.

5.4.3.42 nz0b

```
integer mgrid_mod::nz0b
```

Definition at line 74 of file mgrid_mod.f.

Referenced by becoil(), read_mgrid(), and readin().

5.4.3.43 nzgrid

```
integer mgrid_mod::nzgrid
```

Definition at line 76 of file mgrid_mod.f.

5.4.3.44 pfcspec

```
real(rprec), dimension(:,:), allocatable mgrid_mod::pfcspec
```

Definition at line 96 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.45 plbfld

```
real(rprec), dimension(:,:), allocatable mgrid_mod::plbfld
```

Definition at line 85 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.46 plflux

```
real(rprec), dimension(:), allocatable mgrid_mod::plflux
```

Definition at line 88 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.47 psiext

```
real(rprec), dimension(:), allocatable mgrid_mod::psiext
```

Definition at line 88 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.48 raw_coil_current

```
real(rprec), dimension(:), allocatable mgrid_mod::raw_coil_current
```

Definition at line 87 of file mgrid_mod.f.

Referenced by free_mgrid(), and read_mgrid().

5.4.3.49 rbcoil

```
real(rprec), dimension(:,:), allocatable mgrid_mod::rbcoil
```

Definition at line 85 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.50 rbcoilsqr

```
real(rprec), dimension(:,:), allocatable mgrid_mod::rbcoilsqr
```

Definition at line 85 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.51 reslim

```
real(rprec), dimension(:,:), allocatable mgrid_mod::reslim
```

Definition at line 97 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.52 rlim

```
real(rprec), dimension(:,:), allocatable mgrid_mod::rlim
```

Definition at line 97 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.53 rmaxb

```
real(rprec) mgrid_mod::rmaxb
```

Definition at line 81 of file mgrid_mod.f.

Referenced by becoil(), read_mgrid(), and readin().

5.4.3.54 rminb

```
real(rprec) mgrid_mod::rminb
```

Definition at line 81 of file mgrid_mod.f.

Referenced by becoil(), read_mgrid(), and readin().

5.4.3.55 rx1

```
real(rprec) mgrid_mod::rx1
```

Definition at line 82 of file mgrid_mod.f.

5.4.3.56 rx2

```
real(rprec) mgrid_mod::rx2
```

Definition at line 82 of file mgrid_mod.f.

5.4.3.57 seplim

```
real(rprec), dimension(:,:), allocatable mgrid_mod::seplim
```

Definition at line 97 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.58 tokid

```
character(len=30) mgrid_mod::tokid
```

Definition at line 95 of file mgrid_mod.f.

5.4.3.59 unpsiext

```
real(rprec), dimension(:,:), allocatable mgrid_mod::unpsiext
```

Definition at line 85 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.60 vn_bp0

```
character(len=*), parameter mgrid_mod:vn_bp0 = 'bp'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.61 vn_br0

```
character(len=*), parameter mgrid_mod:vn_br0 = 'br'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.62 vn_bz0

```
character(len=*), parameter mgrid_mod:vn_bz0 = 'bz'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.63 vn_coilcur

```
character(len=*), parameter mgrid_mod:vn_coilcur = 'raw_coil_cur'
```

Definition at line 17 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.64 vn_coilgrp

```
character(len=*), parameter mgrid_mod:vn_coilgrp = 'coil_group'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.65 vn_ir

```
character(len=*), parameter mgrid_mod::vn_ir = 'ir'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.66 vn_jz

```
character(len=*), parameter mgrid_mod::vn_jz = 'jz'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.67 vn_kp

```
character(len=*), parameter mgrid_mod::vn_kp = 'kp'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.68 vn_mgmode

```
character(len=*), parameter mgrid_mod::vn_mgmode = 'mgrid_mode'
```

Definition at line 17 of file mgrid_mod.f.

Referenced by read_mgrid(), read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.4.3.69 vn_nextcur

```
character(len=*), parameter mgrid_mod::vn_nextcur = 'nextcur'
```

Definition at line 17 of file mgrid_mod.f.

Referenced by read_mgrid(), read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.4.3.70 vn_nfp

```
character(len=*), parameter mgrid_mod:vn_nfp = 'nfp'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.71 vn_rmax

```
character(len=*), parameter mgrid_mod:vn_rmax = 'rmax'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.72 vn_rmin

```
character(len=*), parameter mgrid_mod:vn_rmin = 'rmin'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.73 vn_zmax

```
character(len=*), parameter mgrid_mod:vn_zmax = 'zmax'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.74 vn_zmin

```
character(len=*), parameter mgrid_mod:vn_zmin = 'zmin'
```

Definition at line 11 of file mgrid_mod.f.

Referenced by read_mgrid().

5.4.3.75 xobser

```
real(rprec), dimension(:), allocatable mgrid_mod::xobser
```

Definition at line 88 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.76 xobsqr

```
real(rprec), dimension(:), allocatable mgrid_mod::xobsqr
```

Definition at line 88 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.77 zbcoil

```
real(rprec), dimension(:,:), allocatable mgrid_mod::zbcoil
```

Definition at line 85 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.78 zlim

```
real(rprec), dimension(:,:), allocatable mgrid_mod::zlim
```

Definition at line 97 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.79 zmaxb

```
real(rprec) mgrid_mod::zmaxb
```

Definition at line 81 of file mgrid_mod.f.

Referenced by becoil(), read_mgrid(), and readin().

5.4.3.80 zminb

```
real(rprec) mgrid_mod::zminb
```

Definition at line 81 of file mgrid_mod.f.

Referenced by becoil(), read_mgrid(), and readin().

5.4.3.81 zobser

```
real(rprec), dimension(:), allocatable mgrid_mod::zobser
```

Definition at line 88 of file mgrid_mod.f.

Referenced by free_mgrid().

5.4.3.82 zy1

```
real(rprec) mgrid_mod::zy1
```

Definition at line 82 of file mgrid_mod.f.

5.4.3.83 zy2

```
real(rprec) mgrid_mod::zy2
```

Definition at line 82 of file mgrid_mod.f.

5.5 nestor_io Module Reference

Input and Output for stand-alone NESTOR.

Functions/Subroutines

- subroutine [read_nestor_inputs](#) (vac_file)
- subroutine [write_nestor_outputs](#) (vac_file, lasym, ivac, ier_flag)

Variables

- character(len=255) `input_extension`
- character(len=255) `mgrid_file`
- real(dp), dimension(:), allocatable `extcur`
- real(dp), dimension(:), allocatable `raxis`
- real(dp), dimension(:), allocatable `zaxis`
- real(dp), dimension(:), allocatable `xm`
- real(dp), dimension(:), allocatable `xn`
- real(dp), dimension(:), allocatable `rmnc`
- real(dp), dimension(:), allocatable `zmns`
- real(dp), dimension(:), allocatable `rmns`
- real(dp), dimension(:), allocatable `zmnc`
- real(dp), dimension(:), allocatable `wint`
- integer `nfp`
- integer `ntor`
- integer `mpol`
- integer `ntheta`
- integer `nzeta`
- integer `nextcur`
- integer `ier_flag`
- integer `ivac`
- integer `ivacskip`
- integer `mnmax`
- integer `vacuum_calls`
- logical `lasym`
- real(dp) `ctor`
- real(dp) `rbtor`
- real(dp) `signgs`
- integer `mnpd2_nestor`
- real(dp), dimension(:), allocatable `amatsav_nestor`
- real(dp), dimension(:), allocatable `bvecsav_nestor`
- real(dp) `bsubvvac_nestor`
- character(len= *), dimension(1), parameter `mn1dim` = ('mn_mode')
- character(len= *), dimension(1), parameter `mnpotdim` = ('mn_mode_pot')
- character(len= *), dimension(1), parameter `nzntdim` = ('nznt')
- character(len= *), dimension(1), parameter `nzetadim` = ('nzeta')
- character(len= *), dimension(1), parameter `nextcurim` = ('nextcur')
- character(len= *), dimension(1), parameter `bvecsavdim` = ('mnpd2')
- character(len= *), dimension(1), parameter `amatsavdim` = ('mnpd2_times_mnpd2')
- character(len= *), dimension(2), parameter `r2dim` = ('mn_mode','radius')
- character(len= *), parameter `vn_vacuum_calls` = 'vacuum_calls'
- character(len= *), parameter `vn_ier_flag` = "ier_flag"
- character(len= *), parameter `vn_mgrid` = "mgrid_file"
- character(len= *), parameter `vn_inputext` = "input_extension"
- character(len= *), parameter `vn_ivacskip` = "ivacskip"
- character(len= *), parameter `vn_ivac` = "ivac"
- character(len= *), parameter `vn_nfp` = "nfp"
- character(len= *), parameter `vn_ntor` = "ntor"
- character(len= *), parameter `vn_mpol` = "mpol"
- character(len= *), parameter `vn_nzeta` = "nzeta"
- character(len= *), parameter `vn_ntheta` = "ntheta"
- character(len= *), parameter `vn_mnmax` = "mnmax"
- character(len= *), parameter `vn_pmod` = "xm"
- character(len= *), parameter `vn_tmod` = "xn"

- character(len= *), parameter `vn_rmnc` = "rmnc"
- character(len= *), parameter `vn_zmns` = "zmns"
- character(len= *), parameter `vn_rmns` = "rmns"
- character(len= *), parameter `vn_zmnc` = "zmnc"
- character(len= *), parameter `vn_rbtor` = "rbtor"
- character(len= *), parameter `vn_ctor` = "ctor"
- character(len= *), parameter `vn_lasym` = "lasym"
- character(len= *), parameter `vn_signgs` = "signgs"
- character(len= *), parameter `vn_extcur` = "extcur"
- character(len= *), parameter `vn_raxis_nestor` = "raxis_nestor"
- character(len= *), parameter `vn_zaxis_nestor` = "zaxis_nestor"
- character(len= *), parameter `vn_wint` = "wint"
- character(len= *), parameter `vn_bsqvac` = "bsqvac"
- character(len= *), parameter `vn_mnpd` = "mnpd"
- character(len= *), parameter `vn_xmpot` = "xmpot"
- character(len= *), parameter `vn_xnpot` = "xnpot"
- character(len= *), parameter `vn_potvac` = "potvac"
- character(len= *), parameter `vn_brv` = "brv"
- character(len= *), parameter `vn_bphiv` = "bphiv"
- character(len= *), parameter `vn_bzv` = "bzv"
- character(len= *), parameter `vn_bsubvvac` = "bsubvvac"
- character(len= *), parameter `vn_amatsav` = "amatsav"
- character(len= *), parameter `vn_bvecsav` = "bvecsav"
- character(len= *), parameter `vn_mnpd2` = "mnpd2"
- character(len= *), parameter `vn_r1b` = "r1b"
- character(len= *), parameter `vn_rub` = "rub"
- character(len= *), parameter `vn_rvb` = "rvb"
- character(len= *), parameter `vn_z1b` = "z1b"
- character(len= *), parameter `vn_zub` = "zub"
- character(len= *), parameter `vn_zvb` = "zvb"
- character(len= *), parameter `vn_ruu` = "ruu"
- character(len= *), parameter `vn_ruv` = "ruv"
- character(len= *), parameter `vn_rvv` = "rvv"
- character(len= *), parameter `vn_zuu` = "zuu"
- character(len= *), parameter `vn_zuv` = "zuv"
- character(len= *), parameter `vn_zvv` = "zvv"
- character(len= *), parameter `vn_guu_b` = "guu_b"
- character(len= *), parameter `vn_guv_b` = "guv_b"
- character(len= *), parameter `vn_gvv_b` = "gvv_b"
- character(len= *), parameter `vn_rzb2` = "rzb2"
- character(len= *), parameter `vn_snr` = "snr"
- character(len= *), parameter `vn_snv` = "snv"
- character(len= *), parameter `vn_snz` = "snz"
- character(len= *), parameter `vn_drv` = "drv"
- character(len= *), parameter `vn_auu` = "auu"
- character(len= *), parameter `vn_auv` = "auv"
- character(len= *), parameter `vn_avv` = "avv"
- character(len= *), parameter `vn_rcosuv` = "rcosuv"
- character(len= *), parameter `vn_rsinuv` = "rsinuv"
- character(len= *), parameter `vn_brad` = "brad"
- character(len= *), parameter `vn_bphi` = "bphi"
- character(len= *), parameter `vn_bz` = "bz"
- character(len= *), parameter `vn_bexu` = "bexu"
- character(len= *), parameter `vn_bexv` = "bexv"
- character(len= *), parameter `vn_bexn` = "bexn"

- `character(len= *), parameter vn_bexni = "bexni"`
- `character(len= *), parameter vn_grpmn = "grpmn"`
- `character(len= *), parameter vn_adp = "adp"`
- `character(len= *), parameter vn_adm = "adm"`
- `character(len= *), parameter vn_cma = "cma"`
- `character(len= *), parameter vn_sqrtc = "sqrtc"`
- `character(len= *), parameter vn_sqrta = "sqrta"`
- `character(len= *), parameter vn_delt1u = "delt1u"`
- `character(len= *), parameter vn_azp1u = "azp1u"`
- `character(len= *), parameter vn_azm1u = "azm1u"`
- `character(len= *), parameter vn_cma11u = "cma11u"`
- `character(len= *), parameter vn_r1p = "r1p"`
- `character(len= *), parameter vn_r1m = "r1m"`
- `character(len= *), parameter vn_r0p = "r0p"`
- `character(len= *), parameter vn_r0m = "r0m"`
- `character(len= *), parameter vn_ra1p = "ra1p"`
- `character(len= *), parameter vn_ra1m = "ra1m"`
- `character(len= *), parameter vn_sqad1u = "sqad1u"`
- `character(len= *), parameter vn_sqad2u = "sqad2u"`
- `character(len= *), parameter vn_all_tlp = "all_tlp"`
- `character(len= *), parameter vn_all_tlm = "all_tlm"`
- `character(len= *), parameter vn_all_slp = "all_slp"`
- `character(len= *), parameter vn_all_slm = "all_slm"`
- `character(len= *), parameter vn_m_map = "m_map"`
- `character(len= *), parameter vn_n_map = "n_map"`
- `character(len= *), parameter vn_green = "green"`
- `character(len= *), parameter vn_greenp = "greenp"`
- `character(len= *), parameter vn_tanu = "tanu"`
- `character(len= *), parameter vn_tanv = "tanv"`
- `character(len= *), parameter vn_gstore = "gstore"`
- `character(len= *), parameter vn_grpmn_m_map = "grpmn_m_map"`
- `character(len= *), parameter vn_grpmn_n_map = "grpmn_n_map"`
- `character(len= *), parameter vn_imirr = "imirr"`
- `character(len= *), parameter vn_amatrix = "amatrix"`
- `character(len= *), parameter vn_potu = "potu"`
- `character(len= *), parameter vn_potv = "potv"`
- `character(len= *), parameter vn_bsubu = "bsubu"`
- `character(len= *), parameter vn_bsubv = "bsubv"`

5.5.1 Detailed Description

Input and Output for stand-alone NESTOR.

5.5.2 Function/Subroutine Documentation

5.5.2.1 read_nestor_inputs()

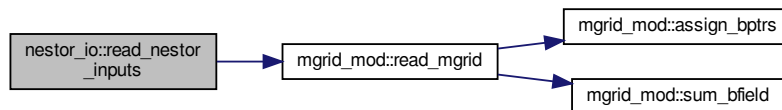
```
subroutine nestor_io::read_nestor_inputs (
    character(len=*), intent(in) vac_file )
```

Definition at line 172 of file nestor_io.f90.

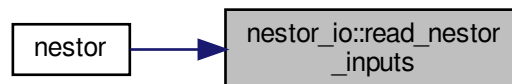
References amatsav_nestor, bsubvvac_nestor, bvecsav_nestor, ctor, stel_kinds::dp, extcur, ier_flag, input_↵ extension, ivac, ivacskip, lasym, mgrid_file, mnmax, mnpd2_nestor, mpol, nfp, ntheta, ntor, nzeta, raxis, rbtor, mgrid_mod::read_mgrid(), rmnc, rmns, signgs, vacuum_calls, vn_amatsav, vn_bsubvvac, vn_bvecsav, vn_ctor, vn_extcur, vn_ier_flag, vn_inputext, vn_ivac, vn_ivacskip, vn_lasym, vn_mgrid, vn_mnmax, vn_mnpd2, vn_mpol, vn_nfp, vn_ntheta, vn_ntor, vn_nzeta, vn_pmod, vn_raxis_nestor, vn_rbtor, vn_rmnc, vn_rmns, vn_signgs, vn_↵ tmod, vn_vacuum_calls, vn_wint, vn_zaxis_nestor, vn_zmnc, vn_zmns, wint, xm, xn, zaxis, zmnc, and zmns.

Referenced by nestor().

Here is the call graph for this function:



Here is the caller graph for this function:



5.5.2.2 write_nestor_outputs()

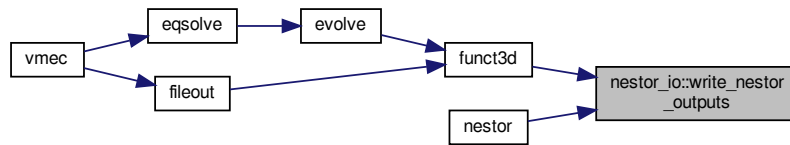
```
subroutine nestor_io::write_nestor_outputs (
    character(len=*), intent(in) vac_file,
    logical, intent(in) lasym,
    integer, intent(in) ivac,
    integer, intent(in) ier_flag )
```

Definition at line 290 of file nestor_io.f90.

References `vacmod::adm`, `vacmod::adp`, `vacmod::all_slm`, `vacmod::all_slp`, `vacmod::all_tlm`, `vacmod::all_tlp`, `vacmod::amatrix`, `vacmod::amatsav`, `vacmod::auu`, `vacmod::auv`, `vacmod::avv`, `vacmod::azm1u`, `vacmod::azp1u`, `vacmod::bexn`, `vacmod::bexni`, `vacmod::bexu`, `vacmod::bexv`, `vacmod::bphi`, `vacmod::bphiv`, `vacmod::brad`, `vacmod::brv`, `vacmod::bsqvac`, `vacmod::bsubu`, `vacmod::bsubv`, `vacmod::bsubvvac`, `vacmod::bvecsav`, `vacmod::bz`, `vacmod::bzv`, `vacmod::cma`, `vacmod::cma11u`, `vacmod::delt1u`, `stel_kinds::dp`, `vacmod::drv`, `vacmod::green`, `vacmod::greenp`, `vacmod::grpmn`, `vacmod::grpmn_m_map_wrt`, `vacmod::grpmn_n_map_wrt`, `vacmod::gstore`, `vacmod::guu_b`, `vacmod::guv_b`, `vacmod::gvv_b`, `vacmod::m_map_wrt`, `vacmod::n_map_wrt`, `vacmod::potu`, `vacmod::potv`, `vacmod::potvac`, `vacmod::r0m`, `vacmod::r0p`, `vacmod::r1b`, `vacmod::r1m`, `vacmod::r1p`, `vacmod::ra1m`, `vacmod::ra1p`, `vacmod::rcosuv`, `vacmod::rsinuv`, `vacmod::rub`, `vacmod::ruu`, `vacmod::ruv`, `vacmod::rvb`, `vacmod::rvv`, `vacmod::rzb2`, `vacmod::snr`, `vacmod::snv`, `vacmod::snz`, `vacmod::sqad1u`, `vacmod::sqad2u`, `vacmod::sqarta`, `vacmod::sqrtc`, `vn_adm`, `vn_adp`, `vn_all_slm`, `vn_all_slp`, `vn_all_tlm`, `vn_all_tlp`, `vn_amatrix`, `vn_amatsav`, `vn_auu`, `vn_auv`, `vn_avv`, `vn_azm1u`, `vn_azp1u`, `vn_bexn`, `vn_bexni`, `vn_bexu`, `vn_bexv`, `vn_bphi`, `vn_bphiv`, `vn_brad`, `vn_brv`, `vn_bsqvac`, `vn_bsubu`, `vn_bsubv`, `vn_bsubvvac`, `vn_bvecsav`, `vn_bz`, `vn_bzv`, `vn_cma`, `vn_cma11u`, `vn_delt1u`, `vn_drv`, `vn_green`, `vn_greenp`, `vn_grpmn`, `vn_grpmn_m_map`, `vn_grpmn_n_map`, `vn_gstore`, `vn_guu_b`, `vn_guv_b`, `vn_gvv_b`, `vn_ier_flag`, `vn_imirr`, `vn_ivac`, `vn_m_map`, `vn_mnpd`, `vn_mnpd2`, `vn_n_map`, `vn_potu`, `vn_potv`, `vn_potvac`, `vn_r0m`, `vn_r0p`, `vn_r1b`, `vn_r1m`, `vn_r1p`, `vn_ra1m`, `vn_ra1p`, `vn_rcosuv`, `vn_rsinuv`, `vn_rub`, `vn_ruu`, `vn_ruv`, `vn_rvb`, `vn_rvv`, `vn_rzb2`, `vn_snr`, `vn_snv`, `vn_snz`, `vn_sqad1u`, `vn_sqad2u`, `vn_sqarta`, `vn_sqrtc`, `vn_tanu`, `vn_tanv`, `vn_xmpot`, `vn_xnpot`, `vn_z1b`, `vn_zub`, `vn_zuu`, `vn_zuv`, `vn_zvb`, `vn_zvv`, `vacmod::z1b`, `vacmod::zub`, `vacmod::zuu`, `vacmod::zuv`, `vacmod::zvb`, and `vacmod::zvv`.

Referenced by `funct3d()`, and `nestor()`.

Here is the caller graph for this function:



5.5.3 Variable Documentation

5.5.3.1 amatsav_nestor

```
real(dp), dimension(:), allocatable nestor_io::amatsav_nestor
```

Definition at line 41 of file `nestor_io.f90`.

Referenced by `nestor()`, and `read_nestor_inputs()`.

5.5.3.2 amatsavdim

```
character(len=*), dimension(1), parameter nestor_io::amatsavdim = ('mnpd2_times_mnpd2'/)
```

Definition at line 51 of file `nestor_io.f90`.

Referenced by `write_nestor_inputs()`.

5.5.3.3 bsubvvac_nestor

```
real(dp) nestor_io::bsubvvac_nestor
```

Definition at line 43 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.4 bvecsav_nestor

```
real(dp), dimension(:), allocatable nestor_io::bvecsav_nestor
```

Definition at line 42 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.5 bvecsavdim

```
character(len=*), dimension(1), parameter nestor_io::bvecsavdim = ('mnpd2'/)
```

Definition at line 50 of file nestor_io.f90.

Referenced by write_nestor_inputs().

5.5.3.6 ctor

```
real(dp) nestor_io::ctor
```

Definition at line 36 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.7 extcur

```
real(dp), dimension(:), allocatable nestor_io::extcur
```

Definition at line 11 of file nestor_io.f90.

Referenced by read_nestor_inputs().

5.5.3.8 ier_flag

```
integer nestor_io::ier_flag
```

Definition at line 28 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.9 input_extension

```
character(len=255) nestor_io::input_extension
```

Definition at line 9 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.10 ivac

```
integer nestor_io::ivac
```

Definition at line 29 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.11 ivacskip

```
integer nestor_io::ivacskip
```

Definition at line 30 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.12 lasym

```
logical nestor_io::lasym
```

Definition at line 34 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.13 mgrid_file

```
character(len=255) nestor_io::mgrid_file
```

Definition at line 10 of file nestor_io.f90.

Referenced by read_nestor_inputs().

5.5.3.14 mn1dim

```
character(len=*), dimension(1), parameter nestor_io::mn1dim = ('mn_mode'/)
```

Definition at line 45 of file nestor_io.f90.

Referenced by write_nestor_inputs().

5.5.3.15 mnmax

```
integer nestor_io::mnmax
```

Definition at line 31 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.16 mnpd2_nestor

```
integer nestor_io::mnpd2_nestor
```

Definition at line 40 of file nestor_io.f90.

Referenced by read_nestor_inputs().

5.5.3.17 mnpotdim

```
character(len=*), dimension(1), parameter nestor_io::mnpotdim = ('mn_mode_pot'/)
```

Definition at line 46 of file nestor_io.f90.

Referenced by write_nestor_inputs().

5.5.3.18 mpol

```
integer nestor_io::mpol
```

Definition at line 24 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.19 nextcur

```
integer nestor_io::nextcur
```

Definition at line 27 of file nestor_io.f90.

5.5.3.20 nextcurim

```
character(len=*), dimension(1), parameter nestor_io::nextcurim = ('nextcur')
```

Definition at line 49 of file nestor_io.f90.

Referenced by write_nestor_inputs().

5.5.3.21 nfp

```
integer nestor_io::nfp
```

Definition at line 22 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.22 ntheta

```
integer nestor_io::ntheta
```

Definition at line 25 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.23 ntor

```
integer nestor_io::ntor
```

Definition at line 23 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.24 nzeta

```
integer nestor_io::nzeta
```

Definition at line 26 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.25 nzetadim

```
character(len=*), dimension(1), parameter nestor_io::nzetadim = ('nzeta')
```

Definition at line 48 of file nestor_io.f90.

Referenced by write_nestor_inputs().

5.5.3.26 nzntdim

```
character(len=*), dimension(1), parameter nestor_io::nzntdim = ('nznt')
```

Definition at line 47 of file nestor_io.f90.

Referenced by write_nestor_inputs().

5.5.3.27 r2dim

```
character(len=*), dimension(2), parameter nestor_io::r2dim = ('mn_mode','radius')
```

Definition at line 53 of file nestor_io.f90.

Referenced by write_nestor_inputs().

5.5.3.28 raxis

```
real(dp), dimension(:), allocatable nestor_io::raxis
```

Definition at line 12 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.29 rbtor

```
real(dp) nestor_io::rbtor
```

Definition at line 37 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.30 rmnc

```
real(dp), dimension(:), allocatable nestor_io::rmnc
```

Definition at line 16 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.31 rmns

```
real(dp), dimension(:), allocatable nestor_io::rmns
```

Definition at line 18 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.32 signgs

```
real(dp) nestor_io::signgs
```

Definition at line 38 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.33 vacuum_calls

```
integer nestor_io::vacuum_calls
```

Definition at line 32 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.34 vn_adm

```
character(len=*), parameter nestor_io::vn_adm = "adm"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.35 vn_adp

```
character(len=*), parameter nestor_io::vn_adp = "adp"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.36 vn_all_slm

```
character(len=*), parameter nestor_io::vn_all_slm = "all_slm"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.37 vn_all_slp

```
character(len=*), parameter nestor_io::vn_all_slp = "all_slp"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.38 vn_all_tlm

```
character(len=*), parameter nestor_io::vn_all_tlm = "all_tlm"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.39 vn_all_tlp

```
character(len=*), parameter nestor_io::vn_all_tlp = "all_tlp"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.40 vn_amatrix

```
character(len=*), parameter nestor_io::vn_amatrix = "amatrix"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.41 vn_amatsav

```
character(len=*), parameter nestor_io::vn_amatsav = "amatsav"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_inputs(), read_nestor_outputs(), write_nestor_inputs(), and write_nestor_outputs().

5.5.3.42 vn_auu

```
character(len=*), parameter nestor_io::vn_auu = "auu"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.43 vn_auv

```
character(len=*), parameter nestor_io::vn_auv = "auv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.44 vn_avv

```
character(len=*), parameter nestor_io::vn_avv = "avv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.45 vn_azm1u

```
character(len=*), parameter nestor_io::vn_azm1u = "azm1u"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.46 vn_azp1u

```
character(len=*), parameter nestor_io::vn_azp1u = "azp1u"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.47 vn_bexn

```
character(len=*), parameter nestor_io::vn_bexn = "bexn"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.48 vn_bexni

```
character(len=*), parameter nestor_io::vn_bexni = "bexni"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.49 vn_bexu

```
character(len=*), parameter nestor_io::vn_bexu = "bexu"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.50 vn_bexv

```
character(len=*), parameter nestor_io::vn_bexv = "bexv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.51 vn_bphi

```
character(len=*), parameter nestor_io::vn_bphi = "bphi"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.52 vn_bphiv

```
character(len=*), parameter nestor_io::vn_bphiv = "bphiv"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_outputs(), and write_nestor_outputs().

5.5.3.53 vn_brad

```
character(len=*), parameter nestor_io::vn_brad = "brad"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.54 vn_brv

```
character(len=*), parameter nestor_io::vn_brv = "brv"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_outputs(), and write_nestor_outputs().

5.5.3.55 vn_bsqvac

```
character(len=*), parameter nestor_io::vn_bsqvac = "bsqvac"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_outputs(), and write_nestor_outputs().

5.5.3.56 vn_bsubu

```
character(len=*), parameter nestor_io::vn_bsubu = "bsubu"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.57 vn_bsubv

```
character(len=*), parameter nestor_io::vn_bsubv = "bsubv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.58 vn_bsubvvac

```
character(len=*), parameter nestor_io::vn_bsubvvac = "bsubvvac"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_inputs(), read_nestor_outputs(), write_nestor_inputs(), and write_nestor_outputs().

5.5.3.59 vn_bvecsav

```
character(len=*), parameter nestor_io::vn_bvecsav = "bvecsav"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_inputs(), read_nestor_outputs(), write_nestor_inputs(), and write_nestor_outputs().

5.5.3.60 vn_bz

```
character(len=*), parameter nestor_io::vn_bz = "bz"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.61 vn_bzv

```
character(len=*), parameter nestor_io::vn_bzv = "bzv"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_outputs(), and write_nestor_outputs().

5.5.3.62 vn_cma

```
character(len=*), parameter nestor_io::vn_cma = "cma"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.63 vn_cma11u

```
character(len=*), parameter nestor_io::vn_cma11u = "cma11u"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.64 vn_ctor

```
character(len=*), parameter nestor_io::vn_ctor = "ctor"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.65 vn_delt1u

```
character(len=*), parameter nestor_io::vn_delt1u = "delt1u"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.66 vn_drv

```
character(len=*), parameter nestor_io::vn_drv = "drv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.67 vn_extcur

```
character(len=*), parameter nestor_io::vn_extcur = "extcur"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.68 vn_green

```
character(len=*), parameter nestor_io::vn_green = "green"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.69 vn_greenp

```
character(len=*), parameter nestor_io::vn_greenp = "greenp"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.70 vn_grpmn

```
character(len=*), parameter nestor_io::vn_grpmn = "grpmn"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.71 vn_grpmn_m_map

```
character(len=*), parameter nestor_io::vn_grpmn_m_map = "grpmn_m_map"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.72 vn_grpmn_n_map

```
character(len=*), parameter nestor_io::vn_grpmn_n_map = "grpmn_n_map"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.73 vn_gstore

```
character(len=*), parameter nestor_io::vn_gstore = "gstore"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.74 vn_guu_b

```
character(len=*), parameter nestor_io::vn_guu_b = "guu_b"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.75 vn_guv_b

```
character(len=*), parameter nestor_io::vn_guv_b = "guv_b"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.76 vn_gvv_b

```
character(len=*), parameter nestor_io::vn_gvv_b = "gvv_b"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.77 vn_ier_flag

```
character(len=*), parameter nestor_io::vn_ier_flag = "ier_flag"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), read_nestor_outputs(), write_nestor_inputs(), and write_nestor_outputs().

5.5.3.78 vn_imirr

```
character(len=*), parameter nestor_io::vn_imirr = "imirr"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.79 vn_inputext

```
character(len=*), parameter nestor_io::vn_inputext = "input_extension"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.80 vn_ivac

```
character(len=*), parameter nestor_io::vn_ivac = "ivac"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), read_nestor_outputs(), write_nestor_inputs(), and write_nestor_outputs().

5.5.3.81 vn_ivacskip

```
character(len=*), parameter nestor_io::vn_ivacskip = "ivacskip"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.82 vn_lasym

```
character(len=*), parameter nestor_io::vn_lasym = "lasym"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.83 vn_m_map

```
character(len=*), parameter nestor_io::vn_m_map = "m_map"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.84 vn_mgrid

```
character(len=*), parameter nestor_io::vn_mgrid = "mgrid_file"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.85 vn_mnmax

```
character(len=*), parameter nestor_io::vn_mnmax = "mnmax"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.86 vn_mnpd

```
character(len=*), parameter nestor_io::vn_mnpd = "mnpd"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_outputs(), and write_nestor_outputs().

5.5.3.87 vn_mnpd2

```
character(len=*), parameter nestor_io::vn_mnpd2 = "mnpd2"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_inputs(), read_nestor_outputs(), write_nestor_inputs(), and write_nestor_outputs().

5.5.3.88 vn_mpol

```
character(len=*), parameter nestor_io::vn_mpol = "mpol"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.89 vn_n_map

```
character(len=*), parameter nestor_io::vn_n_map = "n_map"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.90 vn_nfp

```
character(len=*), parameter nestor_io::vn_nfp = "nfp"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.91 vn_ntheta

```
character(len=*), parameter nestor_io::vn_ntheta = "ntheta"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.92 vn_ntor

```
character(len=*), parameter nestor_io::vn_ntor = "ntor"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.93 vn_nzeta

```
character(len=*), parameter nestor_io::vn_nzeta = "nzeta"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.94 vn_pmod

```
character(len=*), parameter nestor_io::vn_pmod = "xm"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.95 vn_potu

```
character(len=*), parameter nestor_io::vn_potu = "potu"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.96 vn_potv

```
character(len=*), parameter nestor_io::vn_potv = "potv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.97 vn_potvac

```
character(len=*), parameter nestor_io::vn_potvac = "potvac"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_outputs(), and write_nestor_outputs().

5.5.3.98 vn_r0m

```
character(len=*), parameter nestor_io::vn_r0m = "r0m"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.99 vn_r0p

```
character(len=*), parameter nestor_io::vn_r0p = "r0p"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.100 vn_r1b

```
character(len=*), parameter nestor_io::vn_r1b = "r1b"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.101 vn_r1m

```
character(len=*), parameter nestor_io::vn_r1m = "r1m"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.102 vn_r1p

```
character(len=*), parameter nestor_io::vn_r1p = "r1p"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.103 vn_ra1m

```
character(len=*), parameter nestor_io::vn_ra1m = "ra1m"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.104 vn_ra1p

```
character(len=*), parameter nestor_io::vn_ra1p = "ra1p"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.105 vn_raxis_nestor

```
character(len=*), parameter nestor_io::vn_raxis_nestor = "raxis_nestor"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.106 vn_rbtor

```
character(len=*), parameter nestor_io::vn_rbtor = "rbtor"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.107 vn_rcosuv

```
character(len=*), parameter nestor_io::vn_rcosuv = "rcosuv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.108 vn_rmnc

```
character(len=*), parameter nestor_io::vn_rmnc = "rmnc"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.109 vn_rmns

```
character(len=*), parameter nestor_io::vn_rmns = "rmns"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.110 vn_rsinuv

```
character(len=*), parameter nestor_io::vn_rsinuv = "rsinuv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.111 vn_rub

```
character(len=*), parameter nestor_io::vn_rub = "rub"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.112 vn_ruu

```
character(len=*), parameter nestor_io::vn_ruu = "ruu"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.113 vn_ruv

```
character(len=*), parameter nestor_io::vn_ruv = "ruv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.114 vn_rvb

```
character(len=*), parameter nestor_io::vn_rvb = "rvb"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.115 vn_rvv

```
character(len=*), parameter nestor_io::vn_rvv = "rvv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.116 vn_rzb2

```
character(len=*), parameter nestor_io::vn_rzb2 = "rzb2"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.117 vn_signgs

```
character(len=*), parameter nestor_io::vn_signgs = "signgs"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.118 vn_snr

```
character(len=*), parameter nestor_io::vn_snr = "snr"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.119 vn_snv

```
character(len=*), parameter nestor_io::vn_snv = "snv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.120 vn_snz

```
character(len=*), parameter nestor_io::vn_snz = "snz"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.121 vn_sqad1u

```
character(len=*), parameter nestor_io::vn_sqad1u = "sqad1u"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.122 vn_sqad2u

```
character(len=*), parameter nestor_io::vn_sqad2u = "sqad2u"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.123 vn_sqrta

```
character(len=*), parameter nestor_io::vn_sqrta = "sqrta"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.124 vn_sqrta

```
character(len=*), parameter nestor_io::vn_sqrta = "sqrta"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.125 vn_tanu

```
character(len=*), parameter nestor_io::vn_tanu = "tanu"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.126 vn_tanv

```
character(len=*), parameter nestor_io::vn_tanv = "tanv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.127 vn_tmod

```
character(len=*), parameter nestor_io::vn_tmod = "xn"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.128 vn_vacuum_calls

```
character(len=*), parameter nestor_io::vn_vacuum_calls = 'vacuum_calls'
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.129 vn_wint

```
character(len=*), parameter nestor_io::vn_wint = "wint"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.130 vn_xmpot

```
character(len=*), parameter nestor_io::vn_xmpot = "xmpot"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_outputs(), and write_nestor_outputs().

5.5.3.131 vn_xnpot

```
character(len=*), parameter nestor_io::vn_xnpot = "xnpot"
```

Definition at line 83 of file nestor_io.f90.

Referenced by read_nestor_outputs(), and write_nestor_outputs().

5.5.3.132 vn_z1b

```
character(len=*), parameter nestor_io::vn_z1b = "z1b"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.133 vn_zaxis_nestor

```
character(len=*), parameter nestor_io::vn_zaxis_nestor = "zaxis_nestor"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.134 vn_zmnc

```
character(len=*), parameter nestor_io::vn_zmnc = "zmnc"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.135 vn_zmns

```
character(len=*), parameter nestor_io::vn_zmns = "zmns"
```

Definition at line 55 of file nestor_io.f90.

Referenced by read_nestor_inputs(), and write_nestor_inputs().

5.5.3.136 vn_zub

```
character(len=*), parameter nestor_io::vn_zub = "zub"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.137 vn_zuu

```
character(len=*), parameter nestor_io::vn_zuu = "zuu"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.138 vn_zuv

```
character(len=*), parameter nestor_io::vn_zuv = "zuv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.139 vn_zvb

```
character(len=*), parameter nestor_io::vn_zvb = "zvb"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.140 vn_zvv

```
character(len=*), parameter nestor_io::vn_zvv = "zvv"
```

Definition at line 98 of file nestor_io.f90.

Referenced by write_nestor_outputs().

5.5.3.141 wint

```
real(dp), dimension(:), allocatable nestor_io::wint
```

Definition at line 20 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.142 xm

```
real(dp), dimension(:), allocatable nestor_io::xm
```

Definition at line 14 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.143 xn

```
real(dp), dimension(:), allocatable nestor_io::xn
```

Definition at line 15 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.144 zaxis

```
real(dp), dimension(:), allocatable nestor_io::zaxis
```

Definition at line 13 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.145 zmnc

```
real(dp), dimension(:), allocatable nestor_io::zmnc
```

Definition at line 19 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.5.3.146 zmns

```
real(dp), dimension(:), allocatable nestor_io::zmns
```

Definition at line 17 of file nestor_io.f90.

Referenced by nestor(), and read_nestor_inputs().

5.6 read_wout_mod Module Reference

Reading of wout VMEC output file.

Data Types

- interface [read_wout_file](#)

Functions/Subroutines

- subroutine [readw_and_open](#) (file_or_extension, ierr, iopen)
- subroutine [compute_currents](#) (ierror)
- subroutine [read_wout_deallocate](#)
- subroutine [tosuvspace](#) (s_in, u_in, v_in, gsqr, bsupu, bsupv, jsupu, jsupv, lam)
- subroutine [loadrzi](#)

Variables

- character(len= *), parameter [vn_version](#) = 'version_'
- character(len= *), parameter [vn_extension](#) = 'input_extension'
- character(len= *), parameter [vn_mgrid](#) = 'mgrid_file'
- character(len= *), parameter [vn_magen](#) = 'wb'
- character(len= *), parameter [vn_therm](#) = 'wp'
- character(len= *), parameter [vn_gam](#) = 'gamma'
- character(len= *), parameter [vn_maxr](#) = 'rmax_surf'
- character(len= *), parameter [vn_minr](#) = 'rmin_surf'
- character(len= *), parameter [vn_maxz](#) = 'zmax_surf'
- character(len= *), parameter [vn_fp](#) = 'nfp'
- character(len= *), parameter [vn_radnod](#) = 'ns'
- character(len= *), parameter [vn_polmod](#) = 'mpol'
- character(len= *), parameter [vn_tormod](#) = 'ntor'
- character(len= *), parameter [vn_maxmod](#) = 'mnmax'
- character(len= *), parameter [vn_maxit](#) = 'niter'
- character(len= *), parameter [vn_actit](#) = 'itfsq'
- character(len= *), parameter [vn_asym](#) = 'lasym'
- character(len= *), parameter [vn_free](#) = 'lfreeb'
- character(len= *), parameter [vn_error](#) = 'ier_flag'
- character(len= *), parameter [vn_aspect](#) = 'aspect'
- character(len= *), parameter [vn_maxmod_nyq](#) = 'mnmax_nyq'
- character(len= *), parameter [vn_beta](#) = 'betatotal'
- character(len= *), parameter [vn_pbeta](#) = 'betapol'
- character(len= *), parameter [vn_tbeta](#) = 'betator'
- character(len= *), parameter [vn_abeta](#) = 'betaxis'
- character(len= *), parameter [vn_b0](#) = 'b0'
- character(len= *), parameter [vn_rbt0](#) = 'rbtor0'
- character(len= *), parameter [vn_rbt1](#) = 'rbtor'
- character(len= *), parameter [vn_sgs](#) = 'signgs'
- character(len= *), parameter [vn_lar](#) = 'lonLarmor'
- character(len= *), parameter [vn_modb](#) = 'volavgB'
- character(len= *), parameter [vn_ctor](#) = 'ctor'
- character(len= *), parameter [vn_amin](#) = 'Aminor_p'
- character(len= *), parameter [vn_rmaj](#) = 'Rmajor_p'
- character(len= *), parameter [vn_vol](#) = 'volume_p'
- character(len= *), parameter [vn_am](#) = 'am'
- character(len= *), parameter [vn_ai](#) = 'ai'
- character(len= *), parameter [vn_ac](#) = 'ac'
- character(len= *), parameter [vn_ah](#) = 'hot particle fraction'
- character(len= *), parameter [vn_atuname](#) = 'T-perp/T-par'
- character(len= *), parameter [vn_pmass_type](#) = 'pmass_type'
- character(len= *), parameter [vn_piota_type](#) = 'piota_type'
- character(len= *), parameter [vn_pcurr_type](#) = 'pcurr_type'
- character(len= *), parameter [vn_am_aux_s](#) = 'am_aux_s'

- character(len= *), parameter vn_am_aux_f = 'am_aux_f'
- character(len= *), parameter vn_ai_aux_s = 'ai_aux_s'
- character(len= *), parameter vn_ai_aux_f = 'ai_aux_f'
- character(len= *), parameter vn_ac_aux_s = 'ac_aux_s'
- character(len= *), parameter vn_ac_aux_f = 'ac_aux_f'
- character(len= *), parameter vn_mse = 'imse'
- character(len= *), parameter vn_thom = 'itse'
- character(len= *), parameter vn_pmod = 'xm'
- character(len= *), parameter vn_tmod = 'xn'
- character(len= *), parameter vn_pmod_nyq = 'xm_nyq'
- character(len= *), parameter vn_tmod_nyq = 'xn_nyq'
- character(len= *), parameter vn_racc = 'raxis_cc'
- character(len= *), parameter vn_zacs = 'zaxis_cs'
- character(len= *), parameter vn_racs = 'raxis_cs'
- character(len= *), parameter vn_zacc = 'zaxis_cc'
- character(len= *), parameter vn_iotaf = 'iotaf'
- character(len= *), parameter vn_qfact = 'q-factor'
- character(len= *), parameter vn_chi = 'chi'
- character(len= *), parameter vn_chipf = 'chipf'
- character(len= *), parameter vn_presf = 'presf'
- character(len= *), parameter vn_phi = 'phi'
- character(len= *), parameter vn_phipf = 'phipf'
- character(len= *), parameter vn_jcuru = 'jcuru'
- character(len= *), parameter vn_jcurv = 'jcurv'
- character(len= *), parameter vn_iotah = 'iotas'
- character(len= *), parameter vn_mass = 'mass'
- character(len= *), parameter vn_presh = 'pres'
- character(len= *), parameter vn_betah = 'beta_vol'
- character(len= *), parameter vn_buco = 'buco'
- character(len= *), parameter vn_bvco = 'bvco'
- character(len= *), parameter vn_vp = 'vp'
- character(len= *), parameter vn_specw = 'specw'
- character(len= *), parameter vn_phip = 'phips'
- character(len= *), parameter vn_jdotb = 'jdotb'
- character(len= *), parameter vn_overr = 'over_r'
- character(len= *), parameter vn_bgrv = 'bdotgradv'
- character(len= *), parameter vn_merc = 'DMerc'
- character(len= *), parameter vn_mshear = 'DShear'
- character(len= *), parameter vn_mwell = 'DWell'
- character(len= *), parameter vn_mcurr = 'DCurr'
- character(len= *), parameter vn_mgeo = 'DGeod'
- character(len= *), parameter vn_equif = 'equif'
- character(len= *), parameter vn_fsq = 'fsq'
- character(len= *), parameter vn_wdot = 'wdot'
- character(len= *), parameter vn_ftolv = 'ftolv'
- character(len= *), parameter vn_fsqli = 'fsqli'
- character(len= *), parameter vn_fsqr = 'fsqr'
- character(len= *), parameter vn_fsqz = 'fsqz'
- character(len= *), parameter vn_extcur = 'extcur'
- character(len= *), parameter vn_curlab = 'curlabel'
- character(len= *), parameter vn_rmnc = 'rmnc'
- character(len= *), parameter vn_zmns = 'zmns'
- character(len= *), parameter vn_lmns = 'lmns'
- character(len= *), parameter vn_gmnc = 'gmnc'
- character(len= *), parameter vn_bmnc = 'bmnc'

- character(len= *), parameter vn_bsubumnc = 'bsubumnc'
- character(len= *), parameter vn_bsubvmnc = 'bsubvmnc'
- character(len= *), parameter vn_bsubsmns = 'bsubsmns'
- character(len= *), parameter vn_bsupumnc = 'bsupumnc'
- character(len= *), parameter vn_bsupvmnc = 'bsupvmnc'
- character(len= *), parameter vn_rmns = 'rmns'
- character(len= *), parameter vn_zmnc = 'zmnc'
- character(len= *), parameter vn_lmnc = 'lmnc'
- character(len= *), parameter vn_gmns = 'gmns'
- character(len= *), parameter vn_bmns = 'bmns'
- character(len= *), parameter vn_bsubumns = 'bsubumns'
- character(len= *), parameter vn_bsubvmns = 'bsubvmns'
- character(len= *), parameter vn_bsubsmnc = 'bsubsmnc'
- character(len= *), parameter vn_bsupumns = 'bsupumns'
- character(len= *), parameter vn_bsupvmns = 'bsupvmns'
- character(len= *), parameter vn_bsubumnc_sur = 'bsubumnc_sur'
- character(len= *), parameter vn_bsubvmnc_sur = 'bsubvmnc_sur'
- character(len= *), parameter vn_bsupumnc_sur = 'bsupumnc_sur'
- character(len= *), parameter vn_bsupvmnc_sur = 'bsupvmnc_sur'
- character(len= *), parameter vn_bsubumns_sur = 'bsubumns_sur'
- character(len= *), parameter vn_bsubvmns_sur = 'bsubvmns_sur'
- character(len= *), parameter vn_bsupumns_sur = 'bsupumns_sur'
- character(len= *), parameter vn_bsupvmns_sur = 'bsupvmns_sur'
- character(len= *), parameter vn_rbc = 'rbc'
- character(len= *), parameter vn_zbs = 'zbs'
- character(len= *), parameter vn_rbs = 'rbs'
- character(len= *), parameter vn_zbc = 'zbc'
- character(len= *), parameter vn_potvac = 'potvac'
- character(len= *), parameter ln_version = 'VMEC Version'
- character(len= *), parameter ln_extension = 'Input file extension'
- character(len= *), parameter ln_mgrid = 'MGRID file'
- character(len= *), parameter ln_magen = 'Magnetic Energy'
- character(len= *), parameter ln_therm = 'Thermal Energy'
- character(len= *), parameter ln_gam = 'Gamma'
- character(len= *), parameter ln_maxr = 'Maximum R'
- character(len= *), parameter ln_minr = 'Minimum R'
- character(len= *), parameter ln_maxz = 'Maximum Z'
- character(len= *), parameter ln_fp = 'Field Periods'
- character(len= *), parameter ln_radnod = 'Radial nodes'
- character(len= *), parameter ln_polmod = 'Poloidal modes'
- character(len= *), parameter ln_tormod = 'Toroidal modes'
- character(len= *), parameter ln_maxmod = 'Fourier modes'
- character(len= *), parameter ln_maxmod_nyq = 'Fourier modes (Nyquist)'
- character(len= *), parameter ln_maxit = 'Max iterations'
- character(len= *), parameter ln_actit = 'Actual iterations'
- character(len= *), parameter ln_asym = 'Asymmetry'
- character(len= *), parameter ln_recon = 'Reconstruction'
- character(len= *), parameter ln_free = 'Free boundary'
- character(len= *), parameter ln_error = 'Error flag'
- character(len= *), parameter ln_aspect = 'Aspect ratio'
- character(len= *), parameter ln_beta = 'Total beta'
- character(len= *), parameter ln_pbeta = 'Poloidal beta'
- character(len= *), parameter ln_tbeta = 'Toroidal beta'
- character(len= *), parameter ln_abeta = 'Beta axis'
- character(len= *), parameter ln_b0 = 'RB-t over R axis'

- character(len= *), parameter `ln_rbt0` = 'RB-t axis'
- character(len= *), parameter `ln_rbt1` = 'RB-t edge'
- character(len= *), parameter `ln_sgs` = 'Sign jacobian'
- character(len= *), parameter `ln_lar` = 'Ion Larmor radius'
- character(len= *), parameter `ln_modb` = 'avg mod B'
- character(len= *), parameter `ln_ctor` = 'Toroidal current'
- character(len= *), parameter `ln_amin` = 'minor radius'
- character(len= *), parameter `ln_rmaj` = 'major radius'
- character(len= *), parameter `ln_vol` = 'Plasma volume'
- character(len= *), parameter `ln_mse` = 'Number of MSE points'
- character(len= *), parameter `ln_thom` = 'Number of Thompson scattering points'
- character(len= *), parameter `ln_am` = 'Specification parameters for `mass(s)`'
- character(len= *), parameter `ln_ac` = 'Specification parameters for `<J>(s)`'
- character(len= *), parameter `ln_ai` = 'Specification parameters for `iota(s)`'
- character(len= *), parameter `ln_pmass_type` = 'Profile type specifier for `mass(s)`'
- character(len= *), parameter `ln_pcurr_type` = 'Profile type specifier for `<J>(s)`'
- character(len= *), parameter `ln_piota_type` = 'Profile type specifier for `iota(s)`'
- character(len= *), parameter `ln_am_aux_s` = 'Auxiliary-s parameters for `mass(s)`'
- character(len= *), parameter `ln_am_aux_f` = 'Auxiliary-f parameters for `mass(s)`'
- character(len= *), parameter `ln_ac_aux_s` = 'Auxiliary-s parameters for `<J>(s)`'
- character(len= *), parameter `ln_ac_aux_f` = 'Auxiliary-f parameters for `<J>(s)`'
- character(len= *), parameter `ln_ai_aux_s` = 'Auxiliary-s parameters for `iota(s)`'
- character(len= *), parameter `ln_ai_aux_f` = 'Auxiliary-f parameters for `iota(s)`'
- character(len= *), parameter `ln_pmod` = 'Poloidal mode numbers'
- character(len= *), parameter `ln_tmod` = 'Toroidal mode numbers'
- character(len= *), parameter `ln_pmod_nyq` = 'Poloidal mode numbers (Nyquist)'
- character(len= *), parameter `ln_tmod_nyq` = 'Toroidal mode numbers (Nyquist)'
- character(len= *), parameter `ln_racc` = '`raxis` (cosnv)'
- character(len= *), parameter `ln_racs` = '`raxis` (sinnv)'
- character(len= *), parameter `ln_zacs` = '`zaxis` (sinnv)'
- character(len= *), parameter `ln_zacc` = '`zaxis` (cosnv)'
- character(len= *), parameter `ln_iotaf` = 'iota on full mesh'
- character(len= *), parameter `ln_qfact` = 'q-factor on full mesh'
- character(len= *), parameter `ln_presf` = 'pressure on full mesh'
- character(len= *), parameter `ln_phi` = 'Toroidal flux on full mesh'
- character(len= *), parameter `ln_phipf` = '`d(phi)/ds`: Toroidal flux deriv on full mesh'
- character(len= *), parameter `ln_chi` = 'Poloidal flux on full mesh'
- character(len= *), parameter `ln_chipf` = '`d(chi)/ds`: Poroidal flux deriv on full mesh'
- character(len= *), parameter `ln_jcuru` = 'j dot gradu full'
- character(len= *), parameter `ln_jcurv` = 'j dot gradv full'
- character(len= *), parameter `ln_iotah` = 'iota half'
- character(len= *), parameter `ln_mass` = '`mass` half'
- character(len= *), parameter `ln_presh` = 'pressure half'
- character(len= *), parameter `ln_betah` = 'beta half'
- character(len= *), parameter `ln_buco` = 'bsubu half'
- character(len= *), parameter `ln_bvco` = 'bsubv half'
- character(len= *), parameter `ln_vp` = '`volume` deriv half'
- character(len= *), parameter `ln_specw` = 'Spectral width half'
- character(len= *), parameter `ln_phip` = 'tor flux deriv over 2pi half'
- character(len= *), parameter `ln_jdotb` = 'J dot B'
- character(len= *), parameter `ln_bgrv` = 'B dot grad v'
- character(len= *), parameter `ln_merc` = 'Mercier criterion'
- character(len= *), parameter `ln_mshear` = 'Shear Mercier'
- character(len= *), parameter `ln_mwell` = 'Well Mercier'
- character(len= *), parameter `ln_mcurr` = 'Current Mercier'

- character(len= *), parameter [ln_mgeo](#) = 'Geodesic Mercier'
- character(len= *), parameter [ln_equif](#) = 'Average force balance'
- character(len= *), parameter [ln_fsq](#) = 'Residual decay'
- character(len= *), parameter [ln_wdot](#) = 'Wdot decay'
- character(len= *), parameter [ln_extcur](#) = 'External coil currents'
- character(len= *), parameter [ln_fsqr](#) = 'Residual decay - radial'
- character(len= *), parameter [ln_fsqz](#) = 'Residual decay - vertical'
- character(len= *), parameter [ln_fsqli](#) = 'Residual decay - hoop'
- character(len= *), parameter [ln_ftolv](#) = 'Residual decay - requested'
- character(len= *), parameter [ln_curlab](#) = 'External current names'
- character(len= *), parameter [ln_rmnc](#) = 'cosmn component of cylindrical R, full mesh'
- character(len= *), parameter [ln_zmns](#) = 'sinmn component of cylindrical Z, full mesh'
- character(len= *), parameter [ln_lmns](#) = 'sinmn component of lambda, half mesh'
- character(len= *), parameter [ln_gmnc](#) = 'cosmn component of [jacobian](#), half mesh'
- character(len= *), parameter [ln_bmnc](#) = 'cosmn component of mod-B, half mesh'
- character(len= *), parameter [ln_bsubumnc](#) = 'cosmn covariant u-component of B, half mesh'
- character(len= *), parameter [ln_bsubvmnc](#) = 'cosmn covariant v-component of B, half mesh'
- character(len= *), parameter [ln_bsubsmns](#) = 'sinmn covariant s-component of B, full mesh'
- character(len= *), parameter [ln_bsubumnc_sur](#) = 'cosmn bsubu of B, [surface](#)'
- character(len= *), parameter [ln_bsubvmnc_sur](#) = 'cosmn bsubv of B, [surface](#)'
- character(len= *), parameter [ln_bsupumnc_sur](#) = 'cosmn bsupu of B, [surface](#)'
- character(len= *), parameter [ln_bsupvmnc_sur](#) = 'cosmn bsupv of B, [surface](#)'
- character(len= *), parameter [ln_bsupumnc](#) = 'BSUPUmnc half'
- character(len= *), parameter [ln_bsupvmnc](#) = 'BSUPVmnc half'
- character(len= *), parameter [ln_rmns](#) = 'sinmn component of cylindrical R, full mesh'
- character(len= *), parameter [ln_zmnc](#) = 'cosmn component of cylindrical Z, full mesh'
- character(len= *), parameter [ln_lmnc](#) = 'cosmn component of lambda, half mesh'
- character(len= *), parameter [ln_gmns](#) = 'sinmn component of [jacobian](#), half mesh'
- character(len= *), parameter [ln_bmns](#) = 'sinmn component of mod-B, half mesh'
- character(len= *), parameter [ln_bsubumns](#) = 'sinmn covariant u-component of B, half mesh'
- character(len= *), parameter [ln_bsubvmns](#) = 'sinmn covariant v-component of B, half mesh'
- character(len= *), parameter [ln_bsubsmnc](#) = 'cosmn covariant s-component of B, full mesh'
- character(len= *), parameter [ln_bsubumns_sur](#) = 'sinmn bsubu of B, [surface](#)'
- character(len= *), parameter [ln_bsubvmns_sur](#) = 'sinmn bsubv of B, [surface](#)'
- character(len= *), parameter [ln_bsupumns_sur](#) = 'sinmn bsupu of B, [surface](#)'
- character(len= *), parameter [ln_bsupvmns_sur](#) = 'sinmn bsupv of B, [surface](#)'
- character(len= *), parameter [ln_bsupumns](#) = 'BSUPUmns half'
- character(len= *), parameter [ln_bsupvmns](#) = 'BSUPVmns half'
- character(len= *), parameter [ln_rbc](#) = 'Initial boundary R cos(mu-nv) coefficients'
- character(len= *), parameter [ln_zbs](#) = 'Initial boundary Z sin(mu-nv) coefficients'
- character(len= *), parameter [ln_rbs](#) = 'Initial boundary R sin(mu-nv) coefficients'
- character(len= *), parameter [ln_zbc](#) = 'Initial boundary Z cos(mu-nv) coefficients'
- character(len= *), parameter [ln_potvac](#) = 'Vacuum Potential on Boundary'
- integer [nfp](#)
- integer [ns](#)
- integer [mpol](#)
- integer [ntor](#)
- integer [mnmax](#)
- integer [mnmax_nyq](#)
- integer [itfsq](#)
- integer [niter](#)
- integer [iasym](#)
- integer [ierr_vmec](#)
- integer [imse](#)
- integer [itse](#)

- integer [nstore_seq](#)
- integer [isnodes](#)
- integer [ipnodes](#)
- integer [imatch_phiedge](#)
- integer [isigng](#)
- integer [mnyq](#)
- integer [nnyq](#)
- integer [ntmax](#)
- real(rprec) [wb](#)
- real(rprec) [wp](#)
- real(rprec) [gamma](#)
- real(rprec) [pfac](#)
- real(rprec) [rmax_surf](#)
- real(rprec) [rmin_surf](#)
- real(rprec) [zmax_surf](#)
- real(rprec) [aspect](#)
- real(rprec) [betatot](#)
- real(rprec) [betapol](#)
- real(rprec) [betator](#)
- real(rprec) [betaxis](#)
- real(rprec) [b0](#)
- real(rprec) [tswgt](#)
- real(rprec) [msewgt](#)
- real(rprec) [flmwgt](#)
- real(rprec) [bcwgt](#)
- real(rprec) [phidiam](#)
- real(rprec) [version_](#)
- real(rprec) [delphid](#)
- real(rprec) [ionlarmor](#)
- real(rprec) [volavgb](#)
- real(rprec) [fsql](#)
- real(rprec) [fsqr](#)
- real(rprec) [fsqz](#)
- real(rprec) [ftolv](#)
- real(rprec) [aminor](#)
- real(rprec) [rmajor](#)
- real(rprec) [volume](#)
- real(rprec) [rbtor](#)
- real(rprec) [rbtor0](#)
- real(rprec) [itor](#)
- real(rprec) [machsq](#)
- real(rprec), dimension(:,:,:), allocatable [rzi_local](#)
- real(rprec), dimension(:,:), allocatable [rmnc](#)
- real(rprec), dimension(:,:), allocatable [zmns](#)
- real(rprec), dimension(:,:), allocatable [lmns](#)
- real(rprec), dimension(:,:), allocatable [rmns](#)
- real(rprec), dimension(:,:), allocatable [zmnc](#)
- real(rprec), dimension(:,:), allocatable [lmnc](#)
- real(rprec), dimension(:,:), allocatable [bmnc](#)
- real(rprec), dimension(:,:), allocatable [gmnc](#)
- real(rprec), dimension(:,:), allocatable [bsubumnc](#)
- real(rprec), dimension(:,:), allocatable [bsubvmnc](#)
- real(rprec), dimension(:,:), allocatable [bsubsmns](#)
- real(rprec), dimension(:,:), allocatable [bsupumnc](#)
- real(rprec), dimension(:,:), allocatable [bsupvmnc](#)

- `real(rprec), dimension(:, :), allocatable` [curvmnc](#)
- `real(rprec), dimension(:, :), allocatable` [currumnc](#)
- `real(rprec), dimension(:, :), allocatable` [bbc](#)
- `real(rprec), dimension(:, :), allocatable` [raxis](#)
- `real(rprec), dimension(:, :), allocatable` [zaxis](#)
- `real(rprec), dimension(:, :), allocatable` [bmns](#)
- `real(rprec), dimension(:, :), allocatable` [gmns](#)
- `real(rprec), dimension(:, :), allocatable` [bsubumns](#)
- `real(rprec), dimension(:, :), allocatable` [bsubvmns](#)
- `real(rprec), dimension(:, :), allocatable` [bsubsmnc](#)
- `real(rprec), dimension(:, :), allocatable` [bsupumns](#)
- `real(rprec), dimension(:, :), allocatable` [bsupvmns](#)
- `real(rprec), dimension(:, :), allocatable` [currumns](#)
- `real(rprec), dimension(:, :), allocatable` [curvmns](#)
- `real(rprec), dimension(:), allocatable` [iotas](#)
- `real(rprec), dimension(:), allocatable` [iotaf](#)
- `real(rprec), dimension(:), allocatable` [presf](#)
- `real(rprec), dimension(:), allocatable` [phipf](#)
- `real(rprec), dimension(:), allocatable` [mass](#)
- `real(rprec), dimension(:), allocatable` [pres](#)
- `real(rprec), dimension(:), allocatable` [beta_vol](#)
- `real(rprec), dimension(:), allocatable` [xm](#)
- `real(rprec), dimension(:), allocatable` [xn](#)
- `real(rprec), dimension(:), allocatable` [qfact](#)
- `real(rprec), dimension(:), allocatable` [chipf](#)
- `real(rprec), dimension(:), allocatable` [phi](#)
- `real(rprec), dimension(:), allocatable` [chi](#)
- `real(rprec), dimension(:), allocatable` [xm_nyq](#)
- `real(rprec), dimension(:), allocatable` [xn_nyq](#)
- `real(rprec), dimension(:), allocatable` [phip](#)
- `real(rprec), dimension(:), allocatable` [buco](#)
- `real(rprec), dimension(:), allocatable` [bvco](#)
- `real(rprec), dimension(:), allocatable` [vp](#)
- `real(rprec), dimension(:), allocatable` [overr](#)
- `real(rprec), dimension(:), allocatable` [jcuru](#)
- `real(rprec), dimension(:), allocatable` [jcurv](#)
- `real(rprec), dimension(:), allocatable` [specw](#)
- `real(rprec), dimension(:), allocatable` [jdotb](#)
- `real(rprec), dimension(:), allocatable` [bdotgradv](#)
- `real(rprec), dimension(:), allocatable` [fsqt](#)
- `real(rprec), dimension(:), allocatable` [wdot](#)
- `real(rprec), dimension(:), allocatable` [am](#)
- `real(rprec), dimension(:), allocatable` [ac](#)
- `real(rprec), dimension(:), allocatable` [ai](#)
- `real(rprec), dimension(:), allocatable` [am_aux_s](#)
- `real(rprec), dimension(:), allocatable` [am_aux_f](#)
- `real(rprec), dimension(:), allocatable` [ac_aux_s](#)
- `real(rprec), dimension(:), allocatable` [ac_aux_f](#)
- `real(rprec), dimension(:), allocatable` [ai_aux_s](#)
- `real(rprec), dimension(:), allocatable` [ai_aux_f](#)
- `real(rprec), dimension(:), allocatable` [dmerc](#)
- `real(rprec), dimension(:), allocatable` [dshear](#)
- `real(rprec), dimension(:), allocatable` [dwell](#)
- `real(rprec), dimension(:), allocatable` [dcurr](#)
- `real(rprec), dimension(:), allocatable` [dgeod](#)

- real(rprec), dimension(:), allocatable [equip](#)
- real(rprec), dimension(:), allocatable [extcur](#)
- real(rprec), dimension(:), allocatable [sknots](#)
- real(rprec), dimension(:), allocatable [ystark](#)
- real(rprec), dimension(:), allocatable [y2stark](#)
- real(rprec), dimension(:), allocatable [pknots](#)
- real(rprec), dimension(:), allocatable [ythom](#)
- real(rprec), dimension(:), allocatable [y2thom](#)
- real(rprec), dimension(:), allocatable [anglemse](#)
- real(rprec), dimension(:), allocatable [rmid](#)
- real(rprec), dimension(:), allocatable [qmid](#)
- real(rprec), dimension(:), allocatable [shear](#)
- real(rprec), dimension(:), allocatable [presmid](#)
- real(rprec), dimension(:), allocatable [alfa](#)
- real(rprec), dimension(:), allocatable [curmid](#)
- real(rprec), dimension(:), allocatable [rstark](#)
- real(rprec), dimension(:), allocatable [qmeas](#)
- real(rprec), dimension(:), allocatable [datastark](#)
- real(rprec), dimension(:), allocatable [rthom](#)
- real(rprec), dimension(:), allocatable [datathom](#)
- real(rprec), dimension(:), allocatable [dsiobt](#)
- real(rprec), dimension(:), allocatable [potvac](#)
- logical [lasym](#)
- logical [lthreed](#)
- logical [lwout_opened](#) =.false.
- character [mgrid_file](#)
- character [input_extension](#)
- character [pmass_type](#)
- character [pcurr_type](#)
- character [piota_type](#)

5.6.1 Detailed Description

Reading of wout VMEC output file.

5.6.2 Function/Subroutine Documentation

5.6.2.1 compute_currents()

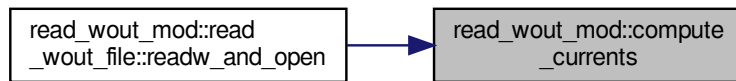
```
subroutine read_wout_mod::compute_currents (
    integer, intent(out) ierror )
```

Definition at line 681 of file read_wout_mod.f.

References [bsubsmnc](#), [bsubsmns](#), [bsubumnc](#), [bsubumns](#), [bsubvmnc](#), [bsubvmns](#), [currumnc](#), [currumns](#), [currvumnc](#), [currvumns](#), [lasym](#), [mnmax_nyq](#), [stel_constants::mu0](#), [xm_nyq](#), and [xn_nyq](#).

Referenced by [read_wout_mod::read_wout_file::readw_and_open\(\)](#).

Here is the caller graph for this function:



5.6.2.2 loadrzl()

```
subroutine read_wout_mod::loadrzl
```

Definition at line 1073 of file `read_wout_mod.f`.

References `chi`, `iotaf`, `lasym`, `lthreed`, `mnmax`, `mpol`, `nfp`, `ns`, `ntmax`, `ntor`, `phipf`, `rmnc`, `rmns`, `rzl_local`, `xm`, `xn`, `zmnc`, and `zmns`.

5.6.2.3 read_wout_deallocate()

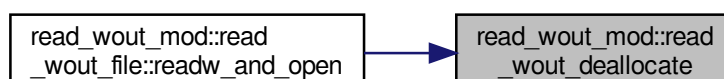
```
subroutine read_wout_mod::read_wout_deallocate
```

Definition at line 781 of file `read_wout_mod.f`.

References `ac`, `ac_aux_f`, `ac_aux_s`, `ai`, `ai_aux_f`, `ai_aux_s`, `am`, `am_aux_f`, `am_aux_s`, `bdotgradv`, `beta_vol`, `bmnc`, `bmns`, `bsubsmnc`, `bsubsmns`, `bsubumnc`, `bsubumns`, `bsubvmnc`, `bsubvmns`, `bsupumnc`, `bsupumns`, `bsupvmnc`, `bsupvmns`, `buco`, `bvco`, `chi`, `chipf`, `mgrid_mod::curlabel`, `currumnc`, `currumns`, `currvmnc`, `currvms`, `dcurr`, `dgeod`, `dmerc`, `dshear`, `dwell`, `equif`, `extcur`, `fsqt`, `gmnc`, `gmns`, `iotaf`, `iotas`, `jcuru`, `jcurv`, `jdotb`, `lmnc`, `lmns`, `lwout_opened`, `mass`, `overr`, `phi`, `phip`, `phipf`, `potvac`, `pres`, `presf`, `raxis`, `rmnc`, `rmns`, `rzl_local`, `specw`, `vp`, `wdot`, `xm`, `xm_nyq`, `xn`, `xn_nyq`, `zaxis`, `zmnc`, and `zmns`.

Referenced by `read_wout_mod::read_wout_file::readw_and_open()`.

Here is the caller graph for this function:



5.6.2.4 readw_and_open()

```
subroutine read_wout_mod::readw_and_open (
    character(len=*), intent(in) file_or_extension,
    integer, intent(out) ierr,
    integer, optional iopen )
```

Definition at line 249 of file read_wout_mod.f.

5.6.2.5 tosuvspace()

```
subroutine read_wout_mod::tosuvspace (
    real(rprec), intent(in) s_in,
    real(rprec), intent(in) u_in,
    real(rprec), intent(in) v_in,
    real(rprec), intent(out), optional gsqrt,
    real(rprec), intent(out), optional bsupu,
    real(rprec), intent(out), optional bsupv,
    real(rprec), intent(out), optional jsupu,
    real(rprec), intent(out), optional jsupv,
    real(rprec), intent(out), optional lam )
```

Definition at line 827 of file read_wout_mod.f.

References bsupumnc, bsupumns, bsupvmnc, bsupvmns, currumnc, currumns, currvmnc, currvmns, gmnc, gmns, lasym, lmnc, lmns, lwout_opened, mnmax_nyq, nfp, nnyq, ns, stel_constants::one, xm_nyq, xn_nyq, and stel_constants::zero.

5.6.3 Variable Documentation

5.6.3.1 ac

```
real(rprec), dimension(:), allocatable read_wout_mod::ac
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.2 ac_aux_f

```
real(rprec), dimension(:), allocatable read_wout_mod::ac_aux_f
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.3 ac_aux_s

```
real(rprec), dimension(:), allocatable read_wout_mod::ac_aux_s
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.4 ai

```
real(rprec), dimension(:), allocatable read_wout_mod::ai
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.5 ai_aux_f

```
real(rprec), dimension(:), allocatable read_wout_mod::ai_aux_f
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.6 ai_aux_s

```
real(rprec), dimension(:), allocatable read_wout_mod::ai_aux_s
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.7 alfa

```
real(rprec), dimension(:), allocatable read_wout_mod::alfa
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.8 am

```
real(rprec), dimension(:), allocatable read_wout_mod::am
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.9 am_aux_f

```
real(rprec), dimension(:), allocatable read_wout_mod::am_aux_f
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.10 am_aux_s

```
real(rprec), dimension(:), allocatable read_wout_mod::am_aux_s
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.11 aminor

```
real(rprec) read_wout_mod::aminor
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.12 anglemse

```
real(rprec), dimension(:), allocatable read_wout_mod::anglemse
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.13 aspect

```
real(rprec) read_wout_mod::aspect
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.14 b0

```
real(rprec) read_wout_mod::b0
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.15 bbc

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bbc
```

Definition at line 212 of file read_wout_mod.f.

5.6.3.16 bcwgt

```
real(rprec) read_wout_mod::bcwgt
```

Definition at line 204 of file read_wout_mod.f.

5.6.3.17 bdotgradv

```
real(rprec), dimension(:), allocatable read_wout_mod::bdotgradv
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.18 beta_vol

```
real(rprec), dimension(:), allocatable read_wout_mod::beta_vol
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.19 betapol

```
real(rprec) read_wout_mod::betapol
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.20 betator

```
real(rprec) read_wout_mod::betator
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.21 betatot

```
real(rprec) read_wout_mod::betatot
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.22 betaxis

```
real(rprec) read_wout_mod::betaxis
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.23 **bmnc**

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bmnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.24 **bmns**

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bmns
```

Definition at line 216 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.25 **bsubsmnc**

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsubsmnc
```

Definition at line 216 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_↵
open().

5.6.3.26 **bsubsmns**

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsubsmns
```

Definition at line 212 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_↵
open().

5.6.3.27 **bsubumnc**

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsubumnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_↵
open().

5.6.3.28 bsubumns

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsubumns
```

Definition at line 216 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_↵
open().

5.6.3.29 bsubvmnc

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsubvmnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_↵
open().

5.6.3.30 bsubvmns

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsubvmns
```

Definition at line 216 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_↵
open().

5.6.3.31 bsupumnc

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsupumnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosubspace().

5.6.3.32 bsupumns

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsupumns
```

Definition at line 216 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosubspace().

5.6.3.33 bsupvmnc

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsupvmnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.34 bsupvmns

```
real(rprec), dimension(:,:), allocatable read_wout_mod::bsupvmns
```

Definition at line 216 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.35 buco

```
real(rprec), dimension(:), allocatable read_wout_mod::buco
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.36 bvco

```
real(rprec), dimension(:), allocatable read_wout_mod::bvco
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.37 chi

```
real(rprec), dimension(:), allocatable read_wout_mod::chi
```

Definition at line 219 of file read_wout_mod.f.

Referenced by loadrzi(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.38 chipf

```
real(rprec), dimension(:), allocatable read_wout_mod::chipf
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.39 curmid

```
real(rprec), dimension(:), allocatable read_wout_mod::curmid
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.40 currumnc

```
real(rprec), dimension(:, :), allocatable read_wout_mod::currumnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and tosuvspace().

5.6.3.41 currumns

```
real(rprec), dimension(:, :), allocatable read_wout_mod::currumns
```

Definition at line 216 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and tosuvspace().

5.6.3.42 currvmnc

```
real(rprec), dimension(:, :), allocatable read_wout_mod::currvmnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and tosuvspace().

5.6.3.43 currvmns

```
real(rprec), dimension(:,:), allocatable read_wout_mod::currvmns
```

Definition at line 216 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), and tosuvspace().

5.6.3.44 datastark

```
real(rprec), dimension(:), allocatable read_wout_mod::datastark
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.45 datathom

```
real(rprec), dimension(:), allocatable read_wout_mod::datathom
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.46 dcurr

```
real(rprec), dimension(:), allocatable read_wout_mod::dcurr
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.47 delphid

```
real(rprec) read_wout_mod::delphid
```

Definition at line 204 of file read_wout_mod.f.

5.6.3.48 dgeod

```
real(rprec), dimension(:), allocatable read_wout_mod::dgeod
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.49 dmerc

```
real(rprec), dimension(:), allocatable read_wout_mod::dmerc
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.50 dshear

```
real(rprec), dimension(:), allocatable read_wout_mod::dshear
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.51 dsiobt

```
real(rprec), dimension(:), allocatable read_wout_mod::dsiobt
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.52 dwell

```
real(rprec), dimension(:), allocatable read_wout_mod::dwell
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.53 equip

```
real(rprec), dimension(:), allocatable read_wout_mod::equip
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.54 extcur

```
real(rprec), dimension(:), allocatable read_wout_mod::extcur
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.55 flmwgt

```
real(rprec) read_wout_mod::flmwgt
```

Definition at line 204 of file read_wout_mod.f.

5.6.3.56 fsql

```
real(rprec) read_wout_mod::fsql
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.57 fsqr

```
real(rprec) read_wout_mod::fsqr
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.58 fsqt

```
real(rprec), dimension(:), allocatable read_wout_mod::fsqt
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.59 fsqz

```
real(rprec) read_wout_mod::fsqz
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.60 ftolv

```
real(rprec) read_wout_mod::ftolv
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.61 gamma

```
real(rprec) read_wout_mod::gamma
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.62 gmnc

```
real(rprec), dimension(:, :), allocatable read_wout_mod::gmnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.63 gmns

```
real(rprec), dimension(:, :), allocatable read_wout_mod::gmns
```

Definition at line 216 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.64 iasym

```
integer read_wout_mod::iasym
```

Definition at line 201 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.65 ierr_vmec

```
integer read_wout_mod::ierr_vmec
```

Definition at line 201 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.66 imatch_phiedge

```
integer read_wout_mod::imatch_phiedge
```

Definition at line 201 of file read_wout_mod.f.

5.6.3.67 imse

```
integer read_wout_mod::imse
```

Definition at line 201 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.68 input_extension

```
character read_wout_mod::input_extension
```

Definition at line 231 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.69 ionlarmor

```
real(rprec) read_wout_mod::ionlarmor
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.70 iotaf

```
real(rprec), dimension(:), allocatable read_wout_mod::iotaf
```

Definition at line 219 of file read_wout_mod.f.

Referenced by loadrzl(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.71 iotas

```
real(rprec), dimension(:), allocatable read_wout_mod::iotas
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.72 ipnodes

```
integer read_wout_mod::ipnodes
```

Definition at line 201 of file read_wout_mod.f.

5.6.3.73 isingn

```
integer read_wout_mod::isingn
```

Definition at line 201 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.74 isnodes

```
integer read_wout_mod::isnodes
```

Definition at line 201 of file read_wout_mod.f.

5.6.3.75 itfsq

```
integer read_wout_mod::itfsq
```

Definition at line 201 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.76 itor

```
real(rprec) read_wout_mod::itor
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.77 itse

```
integer read_wout_mod::itse
```

Definition at line 201 of file read_wout_mod.f.

5.6.3.78 jcuru

```
real(rprec), dimension(:), allocatable read_wout_mod::jcuru
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.79 jcurv

```
real(rprec), dimension(:), allocatable read_wout_mod::jcurv
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.80 jdotb

```
real(rprec), dimension(:), allocatable read_wout_mod::jdotb
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.81 lasym

```
logical read_wout_mod::lasym
```

Definition at line 230 of file read_wout_mod.f.

Referenced by compute_currents(), loadrzi(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.82 lmnc

```
real(rprec), dimension(:,:), allocatable read_wout_mod::lmnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.83 lmns

```
real(rprec), dimension(:,:), allocatable read_wout_mod::lmns
```

Definition at line 212 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.84 `ln_abeta`

```
character(len=*), parameter read_wout_mod::ln_abeta = 'Beta axis'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.85 `ln_ac`

```
character(len=*), parameter read_wout_mod::ln_ac = 'Specification parameters for <J>(s)'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.86 `ln_ac_aux_f`

```
character(len=*), parameter read_wout_mod::ln_ac_aux_f = 'Auxiliary-f parameters for <J>(s)'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.87 `ln_ac_aux_s`

```
character(len=*), parameter read_wout_mod::ln_ac_aux_s = 'Auxiliary-s parameters for <J>(s)'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.88 `ln_actit`

```
character(len=*), parameter read_wout_mod::ln_actit = 'Actual iterations'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.89 ln_ai

```
character(len=*), parameter read_wout_mod::ln_ai = 'Specification parameters for iota(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.90 ln_ai_aux_f

```
character(len=*), parameter read_wout_mod::ln_ai_aux_f = 'Auxiliary-f parameters for iota(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.91 ln_ai_aux_s

```
character(len=*), parameter read_wout_mod::ln_ai_aux_s = 'Auxiliary-s parameters for iota(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.92 ln_am

```
character(len=*), parameter read_wout_mod::ln_am = 'Specification parameters for mass(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.93 ln_am_aux_f

```
character(len=*), parameter read_wout_mod::ln_am_aux_f = 'Auxiliary-f parameters for mass(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.94 ln_am_aux_s

```
character(len=*), parameter read_wout_mod::ln_am_aux_s = 'Auxiliary-s parameters for mass(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.95 ln_amin

```
character(len=*), parameter read_wout_mod::ln_amin = 'minor radius'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.96 ln_aspect

```
character(len=*), parameter read_wout_mod::ln_aspect = 'Aspect ratio'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.97 ln_asym

```
character(len=*), parameter read_wout_mod::ln_asym = 'Asymmetry'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.98 ln_b0

```
character(len=*), parameter read_wout_mod::ln_b0 = 'RB-t over R axis'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.99 ln_beta

```
character(len=*), parameter read_wout_mod::ln_beta = 'Total beta'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.100 ln_betah

```
character(len=*), parameter read_wout_mod::ln_betah = 'beta half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.101 ln_bgrv

```
character(len=*), parameter read_wout_mod::ln_bgrv = 'B dot grad v'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.102 ln_bmnc

```
character(len=*), parameter read_wout_mod::ln_bmnc = 'cosmn component of mod-B, half mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.103 ln_bmns

```
character(len=*), parameter read_wout_mod::ln_bmns = 'sinmn component of mod-B, half mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.104 `ln_bsubsmnc`

```
character(len=*), parameter read_wout_mod::ln_bsubsmnc = 'cosmn covariant s-component of B,  
full mesh'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.105 `ln_bsubsmns`

```
character(len=*), parameter read_wout_mod::ln_bsubsmns = 'sinmn covariant s-component of B,  
full mesh'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.106 `ln_bsubumnc`

```
character(len=*), parameter read_wout_mod::ln_bsubumnc = 'cosmn covariant u-component of B,  
half mesh'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.107 `ln_bsubumnc_sur`

```
character(len=*), parameter read_wout_mod::ln_bsubumnc_sur = 'cosmn bsubu of B, surface'
```

Definition at line 94 of file `read_wout_mod.f`.

5.6.3.108 `ln_bsubumns`

```
character(len=*), parameter read_wout_mod::ln_bsubumns = 'sinmn covariant u-component of B,  
half mesh'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.109 ln_bsubumns_sur

```
character(len=*), parameter read_wout_mod::ln_bsubumns_sur = 'sinmn bsubu of B, surface'
```

Definition at line 94 of file read_wout_mod.f.

5.6.3.110 ln_bsubvmnc

```
character(len=*), parameter read_wout_mod::ln_bsubvmnc = 'cosmn covariant v-component of B,  
half mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.111 ln_bsubvmnc_sur

```
character(len=*), parameter read_wout_mod::ln_bsubvmnc_sur = 'cosmn bsubv of B, surface'
```

Definition at line 94 of file read_wout_mod.f.

5.6.3.112 ln_bsubvmns

```
character(len=*), parameter read_wout_mod::ln_bsubvmns = 'sinmn covariant v-component of B,  
half mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.113 ln_bsubvmns_sur

```
character(len=*), parameter read_wout_mod::ln_bsubvmns_sur = 'sinmn bsubv of B, surface'
```

Definition at line 94 of file read_wout_mod.f.

5.6.3.114 In_bsupumnc

```
character(len=*), parameter read_wout_mod::ln_bsupumnc = 'BSUPUmnc half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.115 In_bsupumnc_sur

```
character(len=*), parameter read_wout_mod::ln_bsupumnc_sur = 'cosmn bsupu of B, surface'
```

Definition at line 94 of file read_wout_mod.f.

5.6.3.116 In_bsupumns

```
character(len=*), parameter read_wout_mod::ln_bsupumns = 'BSUPUmns half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.117 In_bsupumns_sur

```
character(len=*), parameter read_wout_mod::ln_bsupumns_sur = 'sinmn bsupu of B, surface'
```

Definition at line 94 of file read_wout_mod.f.

5.6.3.118 In_bsupvmnc

```
character(len=*), parameter read_wout_mod::ln_bsupvmnc = 'BSUPVmnc half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.119 ln_bsupvmnc_sur

```
character(len=*), parameter read_wout_mod::ln_bsupvmnc_sur = 'cosmn bsupv of B, surface'
```

Definition at line 94 of file read_wout_mod.f.

5.6.3.120 ln_bsupvmns

```
character(len=*), parameter read_wout_mod::ln_bsupvmns = 'BSUPVmns half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.121 ln_bsupvmns_sur

```
character(len=*), parameter read_wout_mod::ln_bsupvmns_sur = 'sinmn bsupv of B, surface'
```

Definition at line 94 of file read_wout_mod.f.

5.6.3.122 ln_buco

```
character(len=*), parameter read_wout_mod::ln_buco = 'bsubu half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.123 ln_bvco

```
character(len=*), parameter read_wout_mod::ln_bvco = 'bsubv half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.124 ln_chi

```
character(len=*), parameter read_wout_mod::ln_chi = 'Poloidal flux on full mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.125 ln_chipf

```
character(len=*), parameter read_wout_mod::ln_chipf = 'd(chi)/ds: Poroidal flux deriv on full mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.126 ln_ctor

```
character(len=*), parameter read_wout_mod::ln_ctor = 'Toroidal current'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.127 ln_curlab

```
character(len=*), parameter read_wout_mod::ln_curlab = 'External current names'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.128 ln_equif

```
character(len=*), parameter read_wout_mod::ln_equif = 'Average force balance'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.129 ln_error

```
character(len=*), parameter read_wout_mod::ln_error = 'Error flag'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.130 ln_extcur

```
character(len=*), parameter read_wout_mod::ln_extcur = 'External coil currents'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.131 ln_extension

```
character(len=*), parameter read_wout_mod::ln_extension = 'Input file extension'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.132 ln_fp

```
character(len=*), parameter read_wout_mod::ln_fp = 'Field Periods'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.133 ln_free

```
character(len=*), parameter read_wout_mod::ln_free = 'Free boundary'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.134 `ln_fsq`

```
character(len=*), parameter read_wout_mod::ln_fsq = 'Residual decay'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.135 `ln_fsqli`

```
character(len=*), parameter read_wout_mod::ln_fsqli = 'Residual decay - hoop'
```

Definition at line 94 of file `read_wout_mod.f`.

5.6.3.136 `ln_fsqr`

```
character(len=*), parameter read_wout_mod::ln_fsqr = 'Residual decay - radial'
```

Definition at line 94 of file `read_wout_mod.f`.

5.6.3.137 `ln_fsqz`

```
character(len=*), parameter read_wout_mod::ln_fsqz = 'Residual decay - vertical'
```

Definition at line 94 of file `read_wout_mod.f`.

5.6.3.138 `ln_ftolv`

```
character(len=*), parameter read_wout_mod::ln_ftolv = 'Residual decay - requested'
```

Definition at line 94 of file `read_wout_mod.f`.

5.6.3.139 `ln_gam`

```
character(len=*), parameter read_wout_mod::ln_gam = 'Gamma'
```

Definition at line 94 of file `read_wout_mod.f`.

Referenced by `wrout()`.

5.6.3.140 ln_gmnc

```
character(len=*), parameter read_wout_mod::ln_gmnc = 'cosmn component of jacobian, half mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.141 ln_gmns

```
character(len=*), parameter read_wout_mod::ln_gmns = 'sinmn component of jacobian, half mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.142 ln_iotaf

```
character(len=*), parameter read_wout_mod::ln_iotaf = 'iota on full mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.143 ln_iotah

```
character(len=*), parameter read_wout_mod::ln_iotah = 'iota half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.144 ln_jcuru

```
character(len=*), parameter read_wout_mod::ln_jcuru = 'j dot gradu full'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.145 In_jcurv

```
character(len=*), parameter read_wout_mod::ln_jcurv = 'j dot gradv full'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.146 In_jdotb

```
character(len=*), parameter read_wout_mod::ln_jdotb = 'J dot B'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.147 In_lar

```
character(len=*), parameter read_wout_mod::ln_lar = 'Ion Larmor radius'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.148 In_lmnc

```
character(len=*), parameter read_wout_mod::ln_lmnc = 'cosmn component of lambda, half mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.149 In_lmns

```
character(len=*), parameter read_wout_mod::ln_lmns = 'sinmn component of lambda, half mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.150 ln_magen

```
character(len=*), parameter read_wout_mod::ln_magen = 'Magnetic Energy'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.151 ln_mass

```
character(len=*), parameter read_wout_mod::ln_mass = 'mass half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.152 ln_maxit

```
character(len=*), parameter read_wout_mod::ln_maxit = 'Max iterations'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.153 ln_maxmod

```
character(len=*), parameter read_wout_mod::ln_maxmod = 'Fourier modes'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.154 ln_maxmod_nyq

```
character(len=*), parameter read_wout_mod::ln_maxmod_nyq = 'Fourier modes (Nyquist)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.155 ln_maxr

```
character(len=*), parameter read_wout_mod::ln_maxr = 'Maximum R'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.156 ln_maxz

```
character(len=*), parameter read_wout_mod::ln_maxz = 'Maximum Z'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.157 ln_mcurr

```
character(len=*), parameter read_wout_mod::ln_mcurr = 'Current Mercier'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.158 ln_merc

```
character(len=*), parameter read_wout_mod::ln_merc = 'Mercier criterion'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.159 ln_mgeo

```
character(len=*), parameter read_wout_mod::ln_mgeo = 'Geodesic Mercier'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.160 ln_mgrid

```
character(len=*), parameter read_wout_mod::ln_mgrid = 'MGRID file'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.161 ln_minr

```
character(len=*), parameter read_wout_mod::ln_minr = 'Minimum R'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.162 ln_modb

```
character(len=*), parameter read_wout_mod::ln_modb = 'avg mod B'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.163 ln_mse

```
character(len=*), parameter read_wout_mod::ln_mse = 'Number of MSE points'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.164 ln_mshear

```
character(len=*), parameter read_wout_mod::ln_mshear = 'Shear Mercier'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.165 ln_mwell

```
character(len=*), parameter read_wout_mod::ln_mwell = 'Well Mercier'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.166 ln_pbeta

```
character(len=*), parameter read_wout_mod::ln_pbeta = 'Poloidal beta'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.167 ln_pcurr_type

```
character(len=*), parameter read_wout_mod::ln_pcurr_type = 'Profile type specifier for <J>(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.168 ln_phi

```
character(len=*), parameter read_wout_mod::ln_phi = 'Toroidal flux on full mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.169 ln_hip

```
character(len=*), parameter read_wout_mod::ln_hip = 'tor flux deriv over 2pi half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.170 ln_phipf

```
character(len=*), parameter read_wout_mod::ln_phipf = 'd(phi)/ds: Toroidal flux deriv on full mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.171 ln_piota_type

```
character(len=*), parameter read_wout_mod::ln_piota_type = 'Profile type specifier for iota(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.172 ln_pmass_type

```
character(len=*), parameter read_wout_mod::ln_pmass_type = 'Profile type specifier for mass(s)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.173 ln_pmod

```
character(len=*), parameter read_wout_mod::ln_pmod = 'Poloidal mode numbers'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.174 ln_pmod_nyq

```
character(len=*), parameter read_wout_mod::ln_pmod_nyq = 'Poloidal mode numbers (Nyquist)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.175 ln_polmod

```
character(len=*), parameter read_wout_mod::ln_polmod = 'Poloidal modes'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.176 ln_potvac

```
character(len=*), parameter read_wout_mod::ln_potvac = 'Vacuum Potential on Boundary'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.177 ln_presf

```
character(len=*), parameter read_wout_mod::ln_presf = 'pressure on full mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.178 ln_presh

```
character(len=*), parameter read_wout_mod::ln_presh = 'pressure half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.179 ln_qfact

```
character(len=*), parameter read_wout_mod::ln_qfact = 'q-factor on full mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.180 ln_racc

```
character(len=*), parameter read_wout_mod::ln_racc = 'raxis (cosnv)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.181 ln_racs

```
character(len=*), parameter read_wout_mod::ln_racs = 'raxis (sinnv)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.182 ln_radnod

```
character(len=*), parameter read_wout_mod::ln_radnod = 'Radial nodes'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.183 ln_rbc

```
character(len=*), parameter read_wout_mod::ln_rbc = 'Initial boundary R cos(mu-nv) coefficients'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.184 ln_rbs

```
character(len=*), parameter read_wout_mod::ln_rbs = 'Initial boundary R sin(mu-nv) coefficients'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.185 ln_rbt0

```
character(len=*), parameter read_wout_mod::ln_rbt0 = 'RB-t axis'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.186 ln_rbt1

```
character(len=*), parameter read_wout_mod::ln_rbt1 = 'RB-t edge'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.187 ln_recon

```
character(len=*), parameter read_wout_mod::ln_recon = 'Reconstruction'
```

Definition at line 94 of file read_wout_mod.f.

5.6.3.188 ln_rmaj

```
character(len=*), parameter read_wout_mod::ln_rmaj = 'major radius'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.189 ln_rmnc

```
character(len=*), parameter read_wout_mod::ln_rmnc = 'cosmn component of cylindrical R, full  
mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.190 ln_rmns

```
character(len=*), parameter read_wout_mod::ln_rmns = 'sinmn component of cylindrical R, full mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.191 ln_sgs

```
character(len=*), parameter read_wout_mod::ln_sgs = 'Sign jacobian'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.192 ln_specw

```
character(len=*), parameter read_wout_mod::ln_specw = 'Spectral width half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.193 ln_tbeta

```
character(len=*), parameter read_wout_mod::ln_tbeta = 'Toroidal beta'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.194 ln_therm

```
character(len=*), parameter read_wout_mod::ln_therm = 'Thermal Energy'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.195 ln_thom

```
character(len=*), parameter read_wout_mod::ln_thom = 'Number of Thompson scattering points'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.196 ln_tmod

```
character(len=*), parameter read_wout_mod::ln_tmod = 'Toroidal mode numbers'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.197 ln_tmod_nyq

```
character(len=*), parameter read_wout_mod::ln_tmod_nyq = 'Toroidal mode numbers (Nyquist)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.198 ln_tormod

```
character(len=*), parameter read_wout_mod::ln_tormod = 'Toroidal modes'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.199 ln_version

```
character(len=*), parameter read_wout_mod::ln_version = 'VMEC Version'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.200 ln_vol

```
character(len=*), parameter read_wout_mod::ln_vol = 'Plasma volume'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.201 ln_vp

```
character(len=*), parameter read_wout_mod::ln_vp = 'volume deriv half'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.202 ln_wdot

```
character(len=*), parameter read_wout_mod::ln_wdot = 'Wdot decay'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.203 ln_zacc

```
character(len=*), parameter read_wout_mod::ln_zacc = 'zaxis (cosnv)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.204 ln_zacs

```
character(len=*), parameter read_wout_mod::ln_zacs = 'zaxis (sinnv)'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.205 ln_zbc

```
character(len=*), parameter read_wout_mod::ln_zbc = 'Initial boundary Z cos(mu-nv) coefficients'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.206 ln_zbs

```
character(len=*), parameter read_wout_mod::ln_zbs = 'Initial boundary Z sin(mu-nv) coefficients'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.207 ln_zmnc

```
character(len=*), parameter read_wout_mod::ln_zmnc = 'cosmn component of cylindrical Z, full  
mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.208 ln_zmns

```
character(len=*), parameter read_wout_mod::ln_zmns = 'sinmn component of cylindrical Z, full  
mesh'
```

Definition at line 94 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.209 lthreed

```
logical read_wout_mod::lthreed
```

Definition at line 230 of file read_wout_mod.f.

Referenced by loadrzl(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.210 lwout_opened

```
logical read_wout_mod::lwout_opened =.false.
```

Definition at line 230 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.211 machsq

```
real(rprec) read_wout_mod::machsq
```

Definition at line 204 of file read_wout_mod.f.

5.6.3.212 mass

```
real(rprec), dimension(:), allocatable read_wout_mod::mass
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.213 mgrid_file

```
character read_wout_mod::mgrid_file
```

Definition at line 231 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.214 mnmax

```
integer read_wout_mod::mnmax
```

Definition at line 201 of file read_wout_mod.f.

Referenced by loadrzl(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.215 mnmax_nyq

```
integer read_wout_mod::mnmax_nyq
```

Definition at line 201 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_mod::read_wout_file::readw_and_open(), and tosubspace().

5.6.3.216 mnyq

```
integer read_wout_mod::mnyq
```

Definition at line 201 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.217 mpol

```
integer read_wout_mod::mpol
```

Definition at line 201 of file read_wout_mod.f.

Referenced by loadrzi(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.218 msewgt

```
real(rprec) read_wout_mod::msewgt
```

Definition at line 204 of file read_wout_mod.f.

5.6.3.219 nfp

```
integer read_wout_mod::nfp
```

Definition at line 201 of file read_wout_mod.f.

Referenced by loadrzi(), read_wout_mod::read_wout_file::readw_and_open(), and tosubspace().

5.6.3.220 niter

```
integer read_wout_mod::niter
```

Definition at line 201 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.221 nnyq

```
integer read_wout_mod::nnyq
```

Definition at line 201 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and tosubspace().

5.6.3.222 ns

```
integer read_wout_mod::ns
```

Definition at line 201 of file read_wout_mod.f.

Referenced by loadrzi(), read_wout_mod::read_wout_file::readw_and_open(), and tosubspace().

5.6.3.223 nstore_seq

```
integer read_wout_mod::nstore_seq
```

Definition at line 201 of file read_wout_mod.f.

5.6.3.224 ntmx

```
integer read_wout_mod::ntmx
```

Definition at line 201 of file read_wout_mod.f.

Referenced by loadrzi().

5.6.3.225 ntor

```
integer read_wout_mod::ntor
```

Definition at line 201 of file read_wout_mod.f.

Referenced by loadrzl(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.226 overr

```
real(rprec), dimension(:), allocatable read_wout_mod::overr
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.227 pcurr_type

```
character read_wout_mod::pcurr_type
```

Definition at line 232 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.228 pfac

```
real(rprec) read_wout_mod::pfac
```

Definition at line 204 of file read_wout_mod.f.

5.6.3.229 phi

```
real(rprec), dimension(:), allocatable read_wout_mod::phi
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.230 phidiam

```
real(rprec) read_wout_mod::phidiam
```

Definition at line 204 of file read_wout_mod.f.

5.6.3.231 phip

```
real(rprec), dimension(:), allocatable read_wout_mod::phip
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.232 phipf

```
real(rprec), dimension(:), allocatable read_wout_mod::phipf
```

Definition at line 219 of file read_wout_mod.f.

Referenced by loadrzi(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.233 piota_type

```
character read_wout_mod::piota_type
```

Definition at line 232 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.234 pknots

```
real(rprec), dimension(:), allocatable read_wout_mod::pknots
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.235 pmass_type

```
character read_wout_mod::pmass_type
```

Definition at line 232 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.236 potvac

```
real(rprec), dimension(:), allocatable read_wout_mod::potvac
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.237 pres

```
real(rprec), dimension(:), allocatable read_wout_mod::pres
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.238 presf

```
real(rprec), dimension(:), allocatable read_wout_mod::presf
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.239 presmid

```
real(rprec), dimension(:), allocatable read_wout_mod::presmid
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.240 qfact

```
real(rprec), dimension(:), allocatable read_wout_mod::qfact
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.241 qmeas

```
real(rprec), dimension(:), allocatable read_wout_mod::qmeas
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.242 qmid

```
real(rprec), dimension(:), allocatable read_wout_mod::qmid
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.243 raxis

```
real(rprec), dimension(:,:), allocatable read_wout_mod::raxis
```

Definition at line 212 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.244 rbtor

```
real(rprec) read_wout_mod::rbtor
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.245 `rbtor0`

```
real(rprec) read_wout_mod::rbtor0
```

Definition at line 204 of file `read_wout_mod.f`.

Referenced by `read_wout_mod::read_wout_file::readw_and_open()`.

5.6.3.246 `rmajor`

```
real(rprec) read_wout_mod::rmajor
```

Definition at line 204 of file `read_wout_mod.f`.

Referenced by `read_wout_mod::read_wout_file::readw_and_open()`.

5.6.3.247 `rmax_surf`

```
real(rprec) read_wout_mod::rmax_surf
```

Definition at line 204 of file `read_wout_mod.f`.

Referenced by `read_wout_mod::read_wout_file::readw_and_open()`.

5.6.3.248 `rmid`

```
real(rprec), dimension(:), allocatable read_wout_mod::rmid
```

Definition at line 219 of file `read_wout_mod.f`.

5.6.3.249 `rmin_surf`

```
real(rprec) read_wout_mod::rmin_surf
```

Definition at line 204 of file `read_wout_mod.f`.

Referenced by `read_wout_mod::read_wout_file::readw_and_open()`.

5.6.3.250 rmnc

```
real(rprec), dimension(:,:), allocatable read_wout_mod::rmnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by loadrzi(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.251 rmns

```
real(rprec), dimension(:,:), allocatable read_wout_mod::rmns
```

Definition at line 212 of file read_wout_mod.f.

Referenced by loadrzi(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.252 rstark

```
real(rprec), dimension(:), allocatable read_wout_mod::rstark
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.253 rthom

```
real(rprec), dimension(:), allocatable read_wout_mod::rthom
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.254 rzl_local

```
real(rprec), dimension(:,:,:), allocatable read_wout_mod::rzl_local
```

Definition at line 211 of file read_wout_mod.f.

Referenced by loadrzi(), and read_wout_deallocate().

5.6.3.255 shear

```
real(rprec), dimension(:), allocatable read_wout_mod::shear
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.256 sknots

```
real(rprec), dimension(:), allocatable read_wout_mod::sknots
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.257 specw

```
real(rprec), dimension(:), allocatable read_wout_mod::specw
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.258 tswgt

```
real(rprec) read_wout_mod::tswgt
```

Definition at line 204 of file read_wout_mod.f.

5.6.3.259 version_

```
real(rprec) read_wout_mod::version_
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.260 vn_abeta

```
character(len=*), parameter read_wout_mod::vn_abeta = 'betaxis'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.261 vn_ac

```
character(len=*), parameter read_wout_mod::vn_ac = 'ac'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.262 vn_ac_aux_f

```
character(len=*), parameter read_wout_mod::vn_ac_aux_f = 'ac_aux_f'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.263 vn_ac_aux_s

```
character(len=*), parameter read_wout_mod::vn_ac_aux_s = 'ac_aux_s'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.264 vn_actit

```
character(len=*), parameter read_wout_mod::vn_actit = 'itfsq'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.265 vn_ah

```
character(len=*), parameter read_wout_mod::vn_ah = 'hot particle fraction'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.266 vn_ai

```
character(len=*), parameter read_wout_mod::vn_ai = 'ai'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.267 vn_ai_aux_f

```
character(len=*), parameter read_wout_mod::vn_ai_aux_f = 'ai_aux_f'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.268 vn_ai_aux_s

```
character(len=*), parameter read_wout_mod::vn_ai_aux_s = 'ai_aux_s'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.269 vn_am

```
character(len=*), parameter read_wout_mod::vn_am = 'am'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.270 vn_am_aux_f

```
character(len=*), parameter read_wout_mod::vn_am_aux_f = 'am_aux_f'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.271 vn_am_aux_s

```
character(len=*), parameter read_wout_mod::vn_am_aux_s = 'am_aux_s'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.272 vn_amin

```
character(len=*), parameter read_wout_mod::vn_amin = 'Aminor_p'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.273 vn_aspect

```
character(len=*), parameter read_wout_mod::vn_aspect = 'aspect'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.274 vn_asym

```
character(len=*), parameter read_wout_mod::vn_asym = 'lasym'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.275 vn_atuname

```
character(len=*), parameter read_wout_mod::vn_atuname = 'T-perp/T-par'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.276 vn_b0

```
character(len=*), parameter read_wout_mod::vn_b0 = 'b0'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.277 vn_beta

```
character(len=*), parameter read_wout_mod::vn_beta = 'betatotal'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.278 vn_betah

```
character(len=*), parameter read_wout_mod::vn_betah = 'beta_vol'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.279 vn_bgrv

```
character(len=*), parameter read_wout_mod::vn_bgrv = 'bdotgradv'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.280 vn_bmnc

```
character(len=*), parameter read_wout_mod::vn_bmnc = 'bmnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.281 vn_bmns

```
character(len=*), parameter read_wout_mod::vn_bmns = 'bmns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.282 vn_bsubsmnc

```
character(len=*), parameter read_wout_mod::vn_bsubsmnc = 'bsubsmnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.283 vn_bsubsmns

```
character(len=*), parameter read_wout_mod::vn_bsubsmns = 'bsubsmns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.284 vn_bsubumnc

```
character(len=*), parameter read_wout_mod::vn_bsubumnc = 'bsubumnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.285 vn_bsubumnc_sur

```
character(len=*), parameter read_wout_mod::vn_bsubumnc_sur = 'bsubumnc_sur'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.286 vn_bsubumns

```
character(len=*), parameter read_wout_mod::vn_bsubumns = 'bsubumns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.287 vn_bsubumns_sur

```
character(len=*), parameter read_wout_mod::vn_bsubumns_sur = 'bsubumns_sur'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.288 vn_bsubvmnc

```
character(len=*), parameter read_wout_mod::vn_bsubvmnc = 'bsubvmnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.289 vn_bsubvmnc_sur

```
character(len=*), parameter read_wout_mod::vn_bsubvmnc_sur = 'bsubvmnc_sur'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.290 vn_bsubvmns

```
character(len=*), parameter read_wout_mod::vn_bsubvmns = 'bsubvmns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.291 vn_bsubvmns_sur

```
character(len=*), parameter read_wout_mod::vn_bsubvmns_sur = 'bsubvmns_sur'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.292 vn_bsupumnc

```
character(len=*), parameter read_wout_mod::vn_bsupumnc = 'bsupumnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.293 vn_bsupumnc_sur

```
character(len=*), parameter read_wout_mod::vn_bsupumnc_sur = 'bsupumnc_sur'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.294 vn_bsupumns

```
character(len=*), parameter read_wout_mod::vn_bsupumns = 'bsupumns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.295 vn_bsupumns_sur

```
character(len=*), parameter read_wout_mod::vn_bsupumns_sur = 'bsupumns_sur'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.296 vn_bsupvmnc

```
character(len=*), parameter read_wout_mod::vn_bsupvmnc = 'bsupvmnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.297 vn_bsupvmnc_sur

```
character(len=*), parameter read_wout_mod::vn_bsupvmnc_sur = 'bsupvmnc_sur'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.298 vn_bsupvmns

```
character(len=*), parameter read_wout_mod::vn_bsupvmns = 'bsupvmns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.299 vn_bsupvmns_sur

```
character(len=*), parameter read_wout_mod::vn_bsupvmns_sur = 'bsupvmns_sur'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.300 vn_buco

```
character(len=*), parameter read_wout_mod::vn_buco = 'buco'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.301 vn_bvco

```
character(len=*), parameter read_wout_mod::vn_bvco = 'bvco'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.302 vn_chi

```
character(len=*), parameter read_wout_mod::vn_chi = 'chi'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.303 vn_chipf

```
character(len=*), parameter read_wout_mod::vn_chipf = 'chipf'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.304 vn_ctor

```
character(len=*), parameter read_wout_mod::vn_ctor = 'ctor'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.305 vn_curlab

```
character(len=*), parameter read_wout_mod::vn_curlab = 'curlabel'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.306 vn_equip

```
character(len=*), parameter read_wout_mod::vn_equip = 'equip'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.307 vn_error

```
character(len=*), parameter read_wout_mod::vn_error = 'ier_flag'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.308 vn_extcur

```
character(len=*), parameter read_wout_mod::vn_extcur = 'extcur'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.309 vn_extension

```
character(len=*), parameter read_wout_mod::vn_extension = 'input_extension'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.310 vn_fp

```
character(len=*), parameter read_wout_mod::vn_fp = 'nfp'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.311 vn_free

```
character(len=*), parameter read_wout_mod::vn_free = 'lfreeb'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.312 vn_fsq

```
character(len=*), parameter read_wout_mod::vn_fsq = 'fsqt'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.313 vn_fsqli

```
character(len=*), parameter read_wout_mod::vn_fsqli = 'fsqli'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.314 vn_fsqr

```
character(len=*), parameter read_wout_mod::vn_fsqr = 'fsqr'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.315 vn_fsqz

```
character(len=*), parameter read_wout_mod::vn_fsqz = 'fsqz'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.316 vn_ftolv

```
character(len=*), parameter read_wout_mod::vn_ftolv = 'ftolv'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.317 vn_gam

```
character(len=*), parameter read_wout_mod::vn_gam = 'gamma'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.318 vn_gmnc

```
character(len=*), parameter read_wout_mod::vn_gmnc = 'gmnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.319 vn_gmns

```
character(len=*), parameter read_wout_mod::vn_gmns = 'gmns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.320 vn_iotaf

```
character(len=*), parameter read_wout_mod::vn_iotaf = 'iotaf'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.321 vn_iotah

```
character(len=*), parameter read_wout_mod::vn_iotah = 'iotas'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.322 vn_jcuru

```
character(len=*), parameter read_wout_mod::vn_jcuru = 'jcuru'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.323 vn_jcurv

```
character(len=*), parameter read_wout_mod::vn_jcurv = 'jcurv'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.324 vn_jdotb

```
character(len=*), parameter read_wout_mod::vn_jdotb = 'jdotb'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.325 vn_lar

```
character(len=*), parameter read_wout_mod::vn_lar = 'IonLarmor'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.326 vn_lmnc

```
character(len=*), parameter read_wout_mod::vn_lmnc = 'lmnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.327 vn_lmns

```
character(len=*), parameter read_wout_mod::vn_lmns = 'lmns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.328 vn_magen

```
character(len=*), parameter read_wout_mod::vn_magen = 'wb'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.329 vn_mass

```
character(len=*), parameter read_wout_mod::vn_mass = 'mass'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.330 vn_maxit

```
character(len=*), parameter read_wout_mod::vn_maxit = 'niter'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.331 vn_maxmod

```
character(len=*), parameter read_wout_mod::vn_maxmod = 'mnmax'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.332 vn_maxmod_nyq

```
character(len=*), parameter read_wout_mod::vn_maxmod_nyq = 'mnmax_nyq'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.333 vn_maxr

```
character(len=*), parameter read_wout_mod::vn_maxr = 'rmax_surf'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.334 vn_maxz

```
character(len=*), parameter read_wout_mod::vn_maxz = 'zmax_surf'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.335 vn_mcurr

```
character(len=*), parameter read_wout_mod::vn_mcurr = 'DCurr'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.336 vn_merc

```
character(len=*), parameter read_wout_mod::vn_merc = 'DMerc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.337 vn_mgeo

```
character(len=*), parameter read_wout_mod::vn_mgeo = 'DGeod'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.338 vn_mgrid

```
character(len=*), parameter read_wout_mod::vn_mgrid = 'mgrid_file'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.339 vn_minr

```
character(len=*), parameter read_wout_mod::vn_minr = 'rmin_surf'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.340 vn_modb

```
character(len=*), parameter read_wout_mod::vn_modb = 'volavgB'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.341 vn_mse

```
character(len=*), parameter read_wout_mod::vn_mse = 'mse'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.342 vn_mshear

```
character(len=*), parameter read_wout_mod::vn_mshear = 'DShear'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.343 vn_mwell

```
character(len=*), parameter read_wout_mod::vn_mwell = 'DWell'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.344 vn_overr

```
character(len=*), parameter read_wout_mod::vn_overr = 'over_r'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.345 vn_pbeta

```
character(len=*), parameter read_wout_mod::vn_pbeta = 'betapol'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.346 vn_pcurr_type

```
character(len=*), parameter read_wout_mod::vn_pcurr_type = 'pcurr_type'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.347 vn_phi

```
character(len=*), parameter read_wout_mod::vn_phi = 'phi'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.348 vn_phip

```
character(len=*), parameter read_wout_mod::vn_phip = 'phips'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.349 vn_phipf

```
character(len=*), parameter read_wout_mod::vn_phipf = 'hiphf'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.350 vn_piota_type

```
character(len=*), parameter read_wout_mod::vn_piota_type = 'piota_type'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.351 vn_pmass_type

```
character(len=*), parameter read_wout_mod::vn_pmass_type = 'pmass_type'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.352 vn_pmod

```
character(len=*), parameter read_wout_mod::vn_pmod = 'xm'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.353 vn_pmod_nyq

```
character(len=*), parameter read_wout_mod::vn_pmod_nyq = 'xm_nyq'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.354 vn_polmod

```
character(len=*), parameter read_wout_mod::vn_polmod = 'mpol'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.355 vn_potvac

```
character(len=*), parameter read_wout_mod::vn_potvac = 'potvac'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.356 vn_presf

```
character(len=*), parameter read_wout_mod::vn_presf = 'presf'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.357 vn_presh

```
character(len=*), parameter read_wout_mod::vn_presh = 'pres'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.358 vn_qfact

```
character(len=*), parameter read_wout_mod::vn_qfact = 'q-factor'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.359 vn_racc

```
character(len=*), parameter read_wout_mod::vn_racc = 'raxis_cc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.360 vn_racs

```
character(len=*), parameter read_wout_mod::vn_racs = 'raxis_cs'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.361 vn_radnod

```
character(len=*), parameter read_wout_mod::vn_radnod = 'ns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.362 vn_rbc

```
character(len=*), parameter read_wout_mod::vn_rbc = 'rbc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.363 vn_rbs

```
character(len=*), parameter read_wout_mod::vn_rbs = 'rbs'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.364 vn_rbt0

```
character(len=*), parameter read_wout_mod::vn_rbt0 = 'rbtor0'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.365 vn_rbt1

```
character(len=*), parameter read_wout_mod::vn_rbt1 = 'rbtor'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.366 vn_rmaj

```
character(len=*), parameter read_wout_mod::vn_rmaj = 'Rmajor_p'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.367 vn_rmnc

```
character(len=*), parameter read_wout_mod::vn_rmnc = 'rmnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.368 vn_rmns

```
character(len=*), parameter read_wout_mod::vn_rmns = 'rmns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.369 vn_sgs

```
character(len=*), parameter read_wout_mod::vn_sgs = 'signgs'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.370 vn_specw

```
character(len=*), parameter read_wout_mod::vn_specw = 'specw'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.371 vn_tbeta

```
character(len=*), parameter read_wout_mod::vn_tbeta = 'betator'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.372 vn_therm

```
character(len=*), parameter read_wout_mod::vn_therm = 'wp'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.373 vn_thom

```
character(len=*), parameter read_wout_mod::vn_thom = 'itse'
```

Definition at line 32 of file read_wout_mod.f.

5.6.3.374 vn_tmod

```
character(len=*), parameter read_wout_mod::vn_tmod = 'xn'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.375 vn_tmod_nyq

```
character(len=*), parameter read_wout_mod::vn_tmod_nyq = 'xn_nyq'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.376 vn_tormod

```
character(len=*), parameter read_wout_mod::vn_tormod = 'ntor'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.377 vn_version

```
character(len=*), parameter read_wout_mod::vn_version = 'version_'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.378 vn_vol

```
character(len=*), parameter read_wout_mod::vn_vol = 'volume_p'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.379 vn_vp

```
character(len=*), parameter read_wout_mod::vn_vp = 'vp'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.380 vn_wdot

```
character(len=*), parameter read_wout_mod::vn_wdot = 'wdot'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.381 vn_zacc

```
character(len=*), parameter read_wout_mod::vn_zacc = 'zaxis_cc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.382 vn_zacs

```
character(len=*), parameter read_wout_mod::vn_zacs = 'zaxis_cs'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.383 vn_zbc

```
character(len=*), parameter read_wout_mod::vn_zbc = 'zbc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.384 vn_zbs

```
character(len=*), parameter read_wout_mod::vn_zbs = 'zbs'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by wrout().

5.6.3.385 vn_zmnc

```
character(len=*), parameter read_wout_mod::vn_zmnc = 'zmnc'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.386 vn_zmns

```
character(len=*), parameter read_wout_mod::vn_zmns = 'zmns'
```

Definition at line 32 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open(), and wrout().

5.6.3.387 volavgb

```
real(rprec) read_wout_mod::volavgb
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.388 volume

```
real(rprec) read_wout_mod::volume
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.389 vp

```
real(rprec), dimension(:), allocatable read_wout_mod::vp
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.390 wb

```
real(rprec) read_wout_mod::wb
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.391 wdot

```
real(rprec), dimension(:), allocatable read_wout_mod::wdot
```

Definition at line 219 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.392 wp

```
real(rprec) read_wout_mod::wp
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.393 xm

```
real(rprec), dimension(:), allocatable read_wout_mod::xm
```

Definition at line 219 of file read_wout_mod.f.

Referenced by loadrzi(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.394 xm_nyq

```
real(rprec), dimension(:), allocatable read_wout_mod::xm_nyq
```

Definition at line 219 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.395 xn

```
real(rprec), dimension(:), allocatable read_wout_mod::xn
```

Definition at line 219 of file read_wout_mod.f.

Referenced by loadrzi(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.396 xn_nyq

```
real(rprec), dimension(:), allocatable read_wout_mod::xn_nyq
```

Definition at line 219 of file read_wout_mod.f.

Referenced by compute_currents(), read_wout_deallocate(), read_wout_mod::read_wout_file::readw_and_open(), and tosuvspace().

5.6.3.397 y2stark

```
real(rprec), dimension(:), allocatable read_wout_mod::y2stark
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.398 y2thom

```
real(rprec), dimension(:), allocatable read_wout_mod::y2thom
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.399 ystark

```
real(rprec), dimension(:), allocatable read_wout_mod::ystark
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.400 ythom

```
real(rprec), dimension(:), allocatable read_wout_mod::ythom
```

Definition at line 219 of file read_wout_mod.f.

5.6.3.401 zaxis

```
real(rprec), dimension(:,:), allocatable read_wout_mod::zaxis
```

Definition at line 212 of file read_wout_mod.f.

Referenced by read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.402 zmax_surf

```
real(rprec) read_wout_mod::zmax_surf
```

Definition at line 204 of file read_wout_mod.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

5.6.3.403 zmnc

```
real(rprec), dimension(:,:), allocatable read_wout_mod::zmnc
```

Definition at line 212 of file read_wout_mod.f.

Referenced by loadrzl(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.6.3.404 zmns

```
real(rprec), dimension(:,:), allocatable read_wout_mod::zmns
```

Definition at line 212 of file read_wout_mod.f.

Referenced by loadrzl(), read_wout_deallocate(), and read_wout_mod::read_wout_file::readw_and_open().

5.7 realspace Module Reference**Variables**

- real(rprec), dimension(:,:), allocatable [r1](#)
- real(rprec), dimension(:,:), allocatable [ru](#)
- real(rprec), dimension(:,:), allocatable [rv](#)
- real(rprec), dimension(:,:), allocatable, target [z1](#)
- real(rprec), dimension(:,:), allocatable [zu](#)
- real(rprec), dimension(:,:), allocatable [zv](#)
- real(rprec), dimension(:,:), allocatable [rcon](#)
- real(rprec), dimension(:,:), allocatable [zcon](#)
- real(rprec), dimension(:), allocatable [guu](#)
- real(rprec), dimension(:), allocatable [guv](#)
- real(rprec), dimension(:), allocatable [gvv](#)
- real(rprec), dimension(:), allocatable [ru0](#)
- real(rprec), dimension(:), allocatable [zu0](#)
- real(rprec), dimension(:), allocatable [gcon](#)
- real(rprec), dimension(:), allocatable [rcon0](#)
- real(rprec), dimension(:), allocatable [zcon0](#)
- real(rprec), dimension(:), allocatable [phip](#)
radial derivative of $\phi/(2\pi)$ on half-grid
- real(rprec), dimension(:), allocatable [chip](#)
radial derivative of $\chi/(2\pi)$ on half-grid
- real(rprec), dimension(:), allocatable [shalf](#)
 \sqrt{s} , two-dimensional array on half-grid
- real(rprec), dimension(:), allocatable [sqrts](#)
 \sqrt{s} , two-dimensional array on full-grid
- real(rprec), dimension(:), allocatable [wint](#)
two-dimensional array for normalizing angle integrations
- real(rprec), dimension(:,:), allocatable, target [extra1](#)
- real(rprec), dimension(:,:), allocatable, target [extra2](#)
- real(rprec), dimension(:,:), allocatable, target [extra3](#)
- real(rprec), dimension(:,:), allocatable, target [extra4](#)

5.7.1 Variable Documentation

5.7.1.1 chip

```
real(rprec), dimension(:), allocatable realspace::chip
```

radial derivative of $\chi/(2\pi)$ on half-grid

Definition at line 26 of file realspace.f90.

Referenced by `add_fluxes()`, `allocate_ns()`, `free_mem_ns()`, and `profil3d()`.

5.7.1.2 extra1

```
real(rprec), dimension(:, :), allocatable, target realspace::extra1
```

Definition at line 31 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.3 extra2

```
real(rprec), dimension(:, :), allocatable, target realspace::extra2
```

Definition at line 32 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.4 extra3

```
real(rprec), dimension(:, :), allocatable, target realspace::extra3
```

Definition at line 33 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.5 extra4

```
real(rprec), dimension(:,:), allocatable, target realspace::extra4
```

Definition at line 34 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.6 gcon

```
real(rprec), dimension(:), allocatable realspace::gcon
```

Definition at line 22 of file realspace.f90.

Referenced by `allocate_funct3d()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.7 guu

```
real(rprec), dimension(:), allocatable realspace::guu
```

Definition at line 17 of file realspace.f90.

Referenced by `add_fluxes()`, `allocate_funct3d()`, `bcovar()`, `forces()`, `free_mem_funct3d()`, and `jxbforce()`.

5.7.1.8 guv

```
real(rprec), dimension(:), allocatable realspace::guv
```

Definition at line 18 of file realspace.f90.

Referenced by `add_fluxes()`, `allocate_funct3d()`, `bcovar()`, `forces()`, `free_mem_funct3d()`, and `jxbforce()`.

5.7.1.9 gvv

```
real(rprec), dimension(:), allocatable realspace::gvv
```

Definition at line 19 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `forces()`, `free_mem_funct3d()`, and `jxbforce()`.

5.7.1.10 phip

```
real(rprec), dimension(:), allocatable realspace::pkip
```

radial derivative of $\phi/(2\pi)$ on half-grid

Definition at line 25 of file realspace.f90.

Referenced by `allocate_ns()`, `eqfor()`, `fileout()`, `free_mem_ns()`, `jxbforce()`, `profil3d()`, and `wrout()`.

5.7.1.11 r1

```
real(rprec), dimension(:, :), allocatable realspace::r1
```

Definition at line 8 of file realspace.f90.

Referenced by `allocate_func3d()`, `aspectratio()`, `bcovar()`, `bss()`, `eqfor()`, `eqsolve()`, `forces()`, `free_mem_func3d()`, `freeb_data()`, `func3d()`, `jacobian()`, and `jxbforce()`.

5.7.1.12 rcon

```
real(rprec), dimension(:, :), allocatable realspace::rcon
```

Definition at line 14 of file realspace.f90.

Referenced by `allocate_func3d()`, `eqfor()`, `fileout()`, `forces()`, `free_mem_func3d()`, and `func3d()`.

5.7.1.13 rcon0

```
real(rprec), dimension(:), allocatable realspace::rcon0
```

Definition at line 23 of file realspace.f90.

Referenced by `allocate_func3d()`, `forces()`, `free_mem_func3d()`, and `func3d()`.

5.7.1.14 ru

```
real(rprec), dimension(:, :), allocatable realspace::ru
```

Definition at line 9 of file realspace.f90.

Referenced by `allocate_func3d()`, `bcovar()`, `forces()`, `free_mem_func3d()`, `func3d()`, `jacobian()`, and `jxbforce()`.

5.7.1.15 ru0

```
real(rprec), dimension(:), allocatable realspace::ru0
```

Definition at line 20 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `eqfor()`, `eqsolve()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.16 rv

```
real(rprec), dimension(:, :), allocatable realspace::rv
```

Definition at line 10 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `bss()`, `eqfor()`, `forces()`, `free_mem_funct3d()`, `funct3d()`, and `jxbforce()`.

5.7.1.17 shalf

```
real(rprec), dimension(:), allocatable realspace::shalf
```

`sqrt(s)` ,two-dimensional array on half-grid

Definition at line 27 of file realspace.f90.

Referenced by `allocate_ns()`, `bcovar()`, `bss()`, `forces()`, `free_mem_ns()`, `jacobian()`, `jxbforce()`, `precondn()`, and `profil1d()`.

5.7.1.18 sqrts

```
real(rprec), dimension(:), allocatable realspace::sqrts
```

`sqrt(s)`, two-dimensional array on full-grid

Definition at line 28 of file realspace.f90.

Referenced by `allocate_ns()`, `bcovar()`, `eqfor()`, `forces()`, `free_mem_ns()`, `funct3d()`, `guess_axis()`, `lamcal()`, `profil1d()`, and `profil3d()`.

5.7.1.19 wint

```
real(rprec), dimension(:), allocatable realspace::wint
```

two-dimensional array for normalizing angle integrations

Definition at line 29 of file realspace.f90.

Referenced by `add_fluxes()`, `allocate_ns()`, `aspectratio()`, `bcovar()`, `calc_fbal()`, `eqfor()`, `free_mem_ns()`, `funct3d()`, `jxbforce()`, `precondn()`, `printout()`, and `profil3d()`.

5.7.1.20 z1

```
real(rprec), dimension(:, :), allocatable, target realspace::z1
```

Definition at line 11 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `bss()`, `eqfor()`, `eqsolve()`, `forces()`, `free_mem_funct3d()`, `freeb_data()`, `funct3d()`, `jacobian()`, and `wrout()`.

5.7.1.21 zcon

```
real(rprec), dimension(:, :), allocatable realspace::zcon
```

Definition at line 15 of file realspace.f90.

Referenced by `allocate_funct3d()`, `eqfor()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.22 zcon0

```
real(rprec), dimension(:), allocatable realspace::zcon0
```

Definition at line 24 of file realspace.f90.

Referenced by `allocate_funct3d()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.23 zu

```
real(rprec), dimension(:, :), allocatable realspace::zu
```

Definition at line 12 of file realspace.f90.

Referenced by `allocate_funct3d()`, `aspectratio()`, `bcovar()`, `forces()`, `free_mem_funct3d()`, `funct3d()`, `jacobian()`, and `jxbforce()`.

5.7.1.24 zu0

```
real(rprec), dimension(:), allocatable realspace::zu0
```

Definition at line 21 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `eqfor()`, `eqsolve()`, `forces()`, `free_mem_funct3d()`, and `funct3d()`.

5.7.1.25 zv

```
real(rprec), dimension(:, :), allocatable realspace::zv
```

Definition at line 13 of file realspace.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `bss()`, `eqfor()`, `forces()`, `free_mem_funct3d()`, `funct3d()`, and `jxbforce()`.

5.8 safe_open_mod Module Reference

Functions/Subroutines

- subroutine `safe_open` (`iunit`, `istat`, `filename`, `filestat`, `fileform`, `record_in`, `access_in`, `delim_in`)

5.8.1 Function/Subroutine Documentation

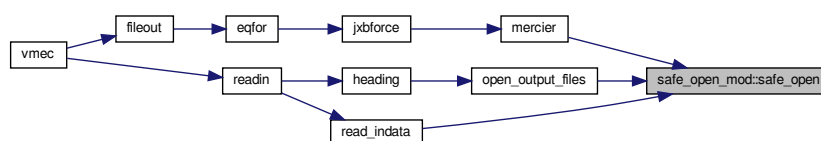
5.8.1.1 safe_open()

```
subroutine safe_open_mod::safe_open (
    integer, intent(inout) iunit,
    integer, intent(out) istat,
    character(len=*), intent(in) filename,
    character(len=*), intent(in) filestat,
    character(len=*), intent(in) fileform,
    integer, intent(in), optional record_in,
    character(len=*), intent(in), optional access_in,
    character(len=*), intent(in), optional delim_in )
```

Definition at line 11 of file `safe_open_mod.f`.

Referenced by `mercier()`, `open_output_files()`, and `read_indata()`.

Here is the caller graph for this function:



5.9 stel_constants Module Reference

Variables

- `real(dp), parameter pi = 3.14159265358979323846264338328_dp`
- `real(dp), parameter pio2 = pi/2`
- `real(dp), parameter twopi = 2*pi`
- `real(dp), parameter sqrt2 = 1.41421356237309504880168872_dp`
- `real(dp), parameter degree = twopi / 360`
- `real(dp), parameter one = 1`
- `real(dp), parameter zero = 0`
- `real(dp), parameter mu0 = 2 * twopi * 1.0e-7_dp`

5.9.1 Variable Documentation

5.9.1.1 degree

```
real(dp), parameter stel_constants::degree = twopi / 360
```

Definition at line 16 of file `stel_constants.f90`.

5.9.1.2 mu0

```
real(dp), parameter stel_constants::mu0 = 2 * twopi * 1.0e-7_dp
```

Definition at line 24 of file `stel_constants.f90`.

Referenced by `read_wout_mod::compute_currents()`, and `read_wout_mod::read_wout_file::readw_and_open()`.

5.9.1.3 one

```
real(dp), parameter stel_constants::one = 1
```

Definition at line 17 of file `stel_constants.f90`.

Referenced by `pcurr()`, `piota()`, `pmass()`, and `read_wout_mod::tosuvspace()`.

5.9.1.4 pi

```
real(dp), parameter stel_constants::pi =3.14159265358979323846264338328_dp
```

Definition at line 12 of file stel_constants.f90.

Referenced by eqfor(), pcurr(), and piota().

5.9.1.5 pio2

```
real(dp), parameter stel_constants::pio2 =pi/2
```

Definition at line 13 of file stel_constants.f90.

5.9.1.6 sqrt2

```
real(dp), parameter stel_constants::sqrt2 =1.41421356237309504880168872_dp
```

Definition at line 15 of file stel_constants.f90.

5.9.1.7 twopi

```
real(dp), parameter stel_constants::twopi =2*pi
```

Definition at line 14 of file stel_constants.f90.

5.9.1.8 zero

```
real(dp), parameter stel_constants::zero =0
```

Definition at line 18 of file stel_constants.f90.

Referenced by line_segment::line_seg_int(), pcurr(), piota(), pmass(), reset_params(), and read_wout_mod↵::tosuvspace().

5.10 stel_kinds Module Reference

Variables

- integer, parameter rprec = SELECTED_REAL_KIND(12, 100)
- integer, parameter iprec = SELECTED_INT_KIND(8)
- integer, parameter cprec = KIND((1.0_rprec, 1.0_rprec))
- integer, parameter dp = rprec

5.10.1 Variable Documentation

5.10.1.1 cprec

```
integer, parameter stel_kinds::cprec = KIND( (1.0_rprec, 1.0_rprec) )
```

Definition at line 12 of file stel_kinds.f90.

5.10.1.2 dp

```
integer, parameter stel_kinds::dp = rprec
```

Definition at line 13 of file stel_kinds.f90.

Referenced by `calc_fbal()`, `mercier()`, `nestor()`, `nestor_io::read_nestor_inputs()`, `read_nestor_outputs()`, `write_nestor_inputs()`, and `nestor_io::write_nestor_outputs()`.

5.10.1.3 iprec

```
integer, parameter stel_kinds::iprec = SELECTED_INT_KIND(8)
```

Definition at line 11 of file stel_kinds.f90.

5.10.1.4 rprec

```
integer, parameter stel_kinds::rprec = SELECTED_REAL_KIND(12, 100)
```

Definition at line 10 of file stel_kinds.f90.

Referenced by `convert_asym()`, `convert_sym()`, `totzspa()`, and `totzsps()`.

5.11 vac_persistent Module Reference

Variables

- integer, dimension(:), allocatable [imirr](#)
- real(rprec), dimension(:), allocatable [sinper](#)
- real(rprec), dimension(:), allocatable [cosper](#)
- real(rprec), dimension(:), allocatable [sinuv](#)
- real(rprec), dimension(:), allocatable [cosuv](#)
- real(rprec), dimension(:), allocatable [tanu](#)
- real(rprec), dimension(:), allocatable [tanv](#)
- real(rprec), dimension(:), allocatable [tanu_1d](#)
- real(rprec), dimension(:), allocatable [tanv_1d](#)
- real(rprec), dimension(:), allocatable [xmpot](#)
- real(rprec), dimension(:), allocatable [xnpot](#)
- real(rprec), dimension(:), allocatable [csign](#)
- real(rprec), dimension(:, :), allocatable [sinu](#)
- real(rprec), dimension(:, :), allocatable [cosu](#)
- real(rprec), dimension(:, :), allocatable [sinv](#)
- real(rprec), dimension(:, :), allocatable [cosv](#)
- real(rprec), dimension(:, :), allocatable [sinui](#)
- real(rprec), dimension(:, :), allocatable [cosui](#)
- real(rprec), dimension(:, :), allocatable [sinu1](#)
- real(rprec), dimension(:, :), allocatable [cosu1](#)
- real(rprec), dimension(:, :), allocatable [sinv1](#)
- real(rprec), dimension(:, :), allocatable [cosv1](#)
- real(rprec), dimension(:, :, :), allocatable [cmns](#)
- real(rprec), dimension(:), allocatable [bsubu_sur](#)
- real(rprec), dimension(:), allocatable [bsubv_sur](#)
- real(rprec), dimension(:), allocatable [bsupu_sur](#)
- real(rprec), dimension(:), allocatable [bsupv_sur](#)

5.11.1 Variable Documentation

5.11.1.1 bsubu_sur

`real(rprec), dimension(:), allocatable vac_persistent::bsubu_sur`

Definition at line 38 of file `vac_persistent.f90`.

5.11.1.2 bsubv_sur

`real(rprec), dimension(:), allocatable vac_persistent::bsubv_sur`

Definition at line 39 of file `vac_persistent.f90`.

5.11.1.3 bsupu_sur

```
real(rprec), dimension(:), allocatable vac_persistent::bsupu_sur
```

Definition at line 40 of file vac_persistent.f90.

5.11.1.4 bsupv_sur

```
real(rprec), dimension(:), allocatable vac_persistent::bsupv_sur
```

Definition at line 41 of file vac_persistent.f90.

5.11.1.5 cmns

```
real(rprec), dimension(:,:,:), allocatable vac_persistent::cmns
```

Definition at line 36 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.6 cosper

```
real(rprec), dimension(:), allocatable vac_persistent::cosper
```

Definition at line 11 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.7 cosu

```
real(rprec), dimension(:,:), allocatable vac_persistent::cosu
```

Definition at line 26 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.8 cosu1

```
real(rprec), dimension(:,:), allocatable vac_persistent::cosu1
```

Definition at line 32 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.9 cosui

```
real(rprec), dimension(:,:), allocatable vac_persistent::cosui
```

Definition at line 30 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.10 cosuv

```
real(rprec), dimension(:), allocatable vac_persistent::cosuv
```

Definition at line 13 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.11 cosv

```
real(rprec), dimension(:,:), allocatable vac_persistent::cosv
```

Definition at line 28 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.12 cosv1

```
real(rprec), dimension(:,:), allocatable vac_persistent::cosv1
```

Definition at line 34 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.13 csign

```
real(rprec), dimension(:), allocatable vac_persistent::csign
```

Definition at line 23 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.14 imirr

```
integer, dimension(:), allocatable vac_persistent::imirr
```

Definition at line 8 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.15 sinper

```
real(rprec), dimension(:), allocatable vac_persistent::sinper
```

Definition at line 10 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.16 sinu

```
real(rprec), dimension(:, :), allocatable vac_persistent::sinu
```

Definition at line 25 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.17 sinu1

```
real(rprec), dimension(:, :), allocatable vac_persistent::sinu1
```

Definition at line 31 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.18 sinui

```
real(rprec), dimension(:,:), allocatable vac_persistent::sinui
```

Definition at line 29 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.19 sinuv

```
real(rprec), dimension(:), allocatable vac_persistent::sinuv
```

Definition at line 12 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.20 sinv

```
real(rprec), dimension(:,:), allocatable vac_persistent::sinv
```

Definition at line 27 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.21 sinv1

```
real(rprec), dimension(:,:), allocatable vac_persistent::sinv1
```

Definition at line 33 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.22 tanu

```
real(rprec), dimension(:), allocatable vac_persistent::tanu
```

Definition at line 15 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.23 tanu_1d

```
real(rprec), dimension(:), allocatable vac_persistent::tanu_1d
```

Definition at line 18 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.24 tanv

```
real(rprec), dimension(:), allocatable vac_persistent::tanv
```

Definition at line 16 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.25 tanv_1d

```
real(rprec), dimension(:), allocatable vac_persistent::tanv_1d
```

Definition at line 19 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.26 xmpot

```
real(rprec), dimension(:), allocatable vac_persistent::xmpot
```

Definition at line 21 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.11.1.27 xnpot

```
real(rprec), dimension(:), allocatable vac_persistent::xnpot
```

Definition at line 22 of file vac_persistent.f90.

Referenced by vacmod::allocate_nestor(), and vacmod::free_mem_nestor().

5.12 vacmod Module Reference

Functions/Subroutines

- subroutine [allocate_nestor](#)
- subroutine [free_mem_nestor](#)

Variables

- real(rprec), parameter [p5](#) = cp5
- real(rprec), parameter [two](#) = c2p0
- real(rprec) [bsubvvac](#)
- real(rprec) [pi2](#)
- real(rprec) [pi3](#)
- real(rprec) [pi4](#)
- real(rprec) [alp](#)
- real(rprec) [alu](#)
- real(rprec) [alv](#)
- real(rprec) [alvp](#)
- real(rprec) [onp](#)
- real(rprec) [onp2](#)
- logical [precal_done](#)
- real(rprec), dimension(:), allocatable, target [potvac](#)
- real(rprec), dimension(:), allocatable [m_map_wrt](#)
- real(rprec), dimension(:), allocatable [n_map_wrt](#)
- real(rprec), dimension(:), allocatable [bvecsav](#)
- real(rprec), dimension(:), allocatable [amatsav](#)
- real(rprec), dimension(:), allocatable [bexni](#)
- real(rprec), dimension(:), allocatable [brv](#)
- real(rprec), dimension(:), allocatable [bphiv](#)
- real(rprec), dimension(:), allocatable [bzv](#)
- real(rprec), dimension(:), allocatable [bsqvac](#)
- real(rprec), dimension(:), allocatable [r1b](#)
- real(rprec), dimension(:), allocatable [rub](#)
- real(rprec), dimension(:), allocatable [rvb](#)
- real(rprec), dimension(:), allocatable [z1b](#)
- real(rprec), dimension(:), allocatable [zub](#)
- real(rprec), dimension(:), allocatable [zvb](#)
- real(rprec), dimension(:), allocatable [bexu](#)
- real(rprec), dimension(:), allocatable [bexv](#)
- real(rprec), dimension(:), allocatable [bexn](#)
- real(rprec), dimension(:), allocatable [auu](#)
- real(rprec), dimension(:), allocatable [auv](#)
- real(rprec), dimension(:), allocatable [avv](#)
- real(rprec), dimension(:), allocatable [snr](#)
- real(rprec), dimension(:), allocatable [snv](#)
- real(rprec), dimension(:), allocatable [snz](#)
- real(rprec), dimension(:), allocatable [drv](#)
- real(rprec), dimension(:), allocatable [guu_b](#)
- real(rprec), dimension(:), allocatable [guv_b](#)
- real(rprec), dimension(:), allocatable [gvv_b](#)
- real(rprec), dimension(:), allocatable [rzb2](#)
- real(rprec), dimension(:), allocatable [rcosuv](#)

- `real(rprec), dimension(:), allocatable` [rsinuv](#)
- `real(rprec), dimension(:), allocatable` [raxis_nestor](#)
- `real(rprec), dimension(:), allocatable` [zaxis_nestor](#)
- `real(rprec), dimension(:), allocatable` [bsubu](#)
- `real(rprec), dimension(:), allocatable` [bsubv](#)
- `real(rprec), dimension(:), allocatable` [potu](#)
- `real(rprec), dimension(:), allocatable` [potv](#)
- `real(rprec), dimension(:), allocatable` [amatrix](#)
- `real(rprec), dimension(:), allocatable` [ruu](#)
- `real(rprec), dimension(:), allocatable` [ruv](#)
- `real(rprec), dimension(:), allocatable` [rvv](#)
- `real(rprec), dimension(:), allocatable` [zuu](#)
- `real(rprec), dimension(:), allocatable` [zuv](#)
- `real(rprec), dimension(:), allocatable` [zvv](#)
- `real(rprec), dimension(:), allocatable` [brad](#)
- `real(rprec), dimension(:), allocatable` [bphi](#)
- `real(rprec), dimension(:), allocatable` [bz](#)
- `real(rprec), dimension(:, :), allocatable` [xpts](#)
- `real(rprec), dimension(:), allocatable` [grpmn](#)
- `real(rprec), dimension(:), allocatable` [grpmn_m_map_wrt](#)
- `real(rprec), dimension(:), allocatable` [grpmn_n_map_wrt](#)
- `real(rprec), dimension(:), allocatable` [gstore](#)
- `real(rprec), dimension(:, :), allocatable` [green](#)
- `real(rprec), dimension(:, :), allocatable` [greenp](#)
- `real(rprec), dimension(:), allocatable` [r0p](#)
- `real(rprec), dimension(:), allocatable` [r1p](#)
- `real(rprec), dimension(:), allocatable` [r0m](#)
- `real(rprec), dimension(:), allocatable` [r1m](#)
- `real(rprec), dimension(:), allocatable` [sqrtc](#)
- `real(rprec), dimension(:), allocatable` [sqrta](#)
- `real(rprec), dimension(:), allocatable` [tlp2](#)
- `real(rprec), dimension(:), allocatable` [tlp1](#)
- `real(rprec), dimension(:), allocatable` [tlp](#)
- `real(rprec), dimension(:), allocatable` [tlm2](#)
- `real(rprec), dimension(:), allocatable` [tlm1](#)
- `real(rprec), dimension(:), allocatable` [tlm](#)
- `real(rprec), dimension(:), allocatable` [adp](#)
- `real(rprec), dimension(:), allocatable` [adm](#)
- `real(rprec), dimension(:), allocatable` [cma](#)
- `real(rprec), dimension(:), allocatable` [ra1p](#)
- `real(rprec), dimension(:), allocatable` [ra1m](#)
- `real(rprec), dimension(:), allocatable` [slm](#)
- `real(rprec), dimension(:), allocatable` [slp](#)
- `real(rprec), dimension(:), allocatable` [tlpm](#)
- `real(rprec), dimension(:), allocatable` [slpm](#)
- `real(rprec), dimension(:), allocatable` [delt1u](#)
- `real(rprec), dimension(:), allocatable` [azp1u](#)
- `real(rprec), dimension(:), allocatable` [azm1u](#)
- `real(rprec), dimension(:), allocatable` [cma11u](#)
- `real(rprec), dimension(:), allocatable` [sqad1u](#)
- `real(rprec), dimension(:), allocatable` [sqad2u](#)
- `real(rprec), dimension(:, :), allocatable` [all_tlp](#)
- `real(rprec), dimension(:, :), allocatable` [all_tlm](#)
- `real(rprec), dimension(:, :), allocatable` [all_slp](#)
- `real(rprec), dimension(:, :), allocatable` [all_slm](#)

- real(rprec), dimension(:), allocatable [gsave](#)
- real(rprec), dimension(:), allocatable [ga1](#)
- real(rprec), dimension(:), allocatable [ga2](#)
- real(rprec), dimension(:), allocatable [dsave](#)
- real(rprec), dimension(:, :, :), allocatable [g1](#)
- real(rprec), dimension(:, :, :), allocatable [g2](#)
- real(rprec), dimension(:, :, :), allocatable [bcos](#)
- real(rprec), dimension(:, :, :), allocatable [bsin](#)
- real(rprec), dimension(:, :, :), allocatable [source](#)
- real(rprec), dimension(:, :, :, :), allocatable [actemp](#)
- real(rprec), dimension(:, :, :, :), allocatable [astemp](#)

5.12.1 Function/Subroutine Documentation

5.12.1.1 allocate_nestor()

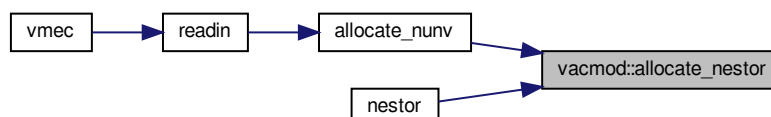
```
subroutine vacmod::allocate_nestor
```

Definition at line 161 of file vacmod.f90.

References [actemp](#), [adm](#), [adp](#), [all_slm](#), [all_slp](#), [all_tlm](#), [all_tlp](#), [amatrix](#), [amatsav](#), [astemp](#), [auu](#), [auv](#), [avv](#), [azm1u](#), [azp1u](#), [bcos](#), [bexn](#), [bexni](#), [bexu](#), [bexv](#), [bphi](#), [bphiv](#), [brad](#), [brv](#), [bsin](#), [bsqvac](#), [bsubu](#), [bsubv](#), [bvecsav](#), [bz](#), [bzv](#), [cma](#), [cma11u](#), [vac_persistent::cmns](#), [vac_persistent::cosper](#), [vac_persistent::cosu](#), [vac_persistent::cosu1](#), [vac_persistent::cosui](#), [vac_persistent::cosuv](#), [vac_persistent::cosv](#), [vac_persistent::cosv1](#), [vac_persistent::csign](#), [delt1u](#), [drv](#), [dsave](#), [g1](#), [g2](#), [ga1](#), [ga2](#), [green](#), [greenp](#), [grpmn](#), [grpmn_m_map_wrt](#), [grpmn_n_map_wrt](#), [gsave](#), [gstore](#), [guu_b](#), [guv_b](#), [gvv_b](#), [vac_persistent::imirr](#), [m_map_wrt](#), [vacmod0::mf](#), [vacmod0::mnpd](#), [vacmod0::mnpd2](#), [n_map_wrt](#), [vacmod0::ndim](#), [vacmod0::nf](#), [vacmod0::nu](#), [vacmod0::nu2](#), [vacmod0::nu3](#), [vacmod0::nuv](#), [vacmod0::nuv2](#), [vacmod0::nuv_tan](#), [vacmod0::nv](#), [vacmod0::nvp](#), [vacmod0::nvper](#), [potu](#), [potv](#), [potvac](#), [precal_done](#), [r0m](#), [r0p](#), [r1b](#), [r1m](#), [r1p](#), [ra1m](#), [ra1p](#), [raxis_nestor](#), [rcosuv](#), [rsinuv](#), [rub](#), [ruu](#), [ruv](#), [rvb](#), [rvv](#), [rzb2](#), [vac_persistent::sinper](#), [vac_persistent::sinu](#), [vac_persistent::sinu1](#), [vac_persistent::sinui](#), [vac_persistent::sinuv](#), [vac_persistent::sinv](#), [vac_persistent::sinv1](#), [slm](#), [slp](#), [slpm](#), [snr](#), [snv](#), [snz](#), [source](#), [sqad1u](#), [sqad2u](#), [sqarta](#), [sqrtc](#), [vac_persistent::tanu](#), [vac_persistent::tanu_1d](#), [vac_persistent::tanv](#), [vac_persistent::tanv_1d](#), [tlm](#), [tlm1](#), [tlm2](#), [tlp](#), [tlp1](#), [tlp2](#), [tlpm](#), [vac_persistent::xmpot](#), [vac_persistent::xnpot](#), [xpts](#), [z1b](#), [zaxis_nestor](#), [zub](#), [zuu](#), [zuv](#), [zvb](#), and [zvv](#).

Referenced by [allocate_nunv\(\)](#), and [nestor\(\)](#).

Here is the caller graph for this function:



5.12.1.2 free_mem_nestor()

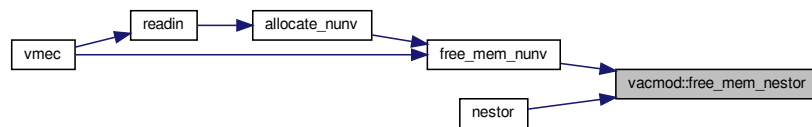
```
subroutine vacmod::free_mem_nestor
```

Definition at line 282 of file vacmod.f90.

References actemp, adm, adp, amatrix, amatsav, astemp, auu, auv, avv, azm1u, azp1u, bcos, bexn, bexni, bexu, bexv, bphi, bphiv, brad, brv, bsin, bsqvac, bsubu, bsubv, bvecsav, bz, bzv, cma, cma11u, vac_persistent::cmns, vac_persistent::cosper, vac_persistent::cosu, vac_persistent::cosu1, vac_persistent::cosui, vac_persistent::cosuv, vac_persistent::cosv, vac_persistent::cosv1, vac_persistent::csign, delt1u, drv, dsave, g1, g2, ga1, ga2, green, greenp, grpmn, grpmn_m_map_wrt, grpmn_n_map_wrt, gsave, gstore, guu_b, guv_b, gvv_b, vac_persistent::imirr, m_map_wrt, n_map_wrt, potu, potv, potvac, r0m, r0p, r1b, r1m, r1p, ra1m, ra1p, raxis_nestor, rcosuv, rsinuv, rub, ruu, ruv, rvb, rvv, rzb2, vac_persistent::sinper, vac_persistent::sinu, vac_persistent::sinu1, vac_persistent::sinui, vac_persistent::sinuv, vac_persistent::sinv, vac_persistent::sinv1, slm, slp, slpm, snr, snv, snz, source, squad1u, squad2u, sqrta, sqrtc, vac_persistent::tanu, vac_persistent::tanu_1d, vac_persistent::tanv, vac_persistent::tanv_1d, tlm, tlm1, tlm2, tlp, tlp1, tlp2, tlpn, vac_persistent::xmpot, vac_persistent::xnpot, xpts, z1b, zaxis_nestor, zub, zuu, zuv, zvb, and zvv.

Referenced by free_mem_nunv(), and nestor().

Here is the caller graph for this function:



5.12.2 Variable Documentation

5.12.2.1 actemp

```
real(rprec), dimension(:,:,:), allocatable vacmod::actemp
```

Definition at line 156 of file vacmod.f90.

Referenced by allocate_nestor(), fouri(), and free_mem_nestor().

5.12.2.2 adm

```
real(rprec), dimension(:), allocatable vacmod::adm
```

Definition at line 121 of file vacmod.f90.

Referenced by allocate_nestor(), analyt(), free_mem_nestor(), and nestor_io::write_nestor_outputs().

5.12.2.3 adp

```
real(rprec), dimension(:), allocatable vacmod::adp
```

Definition at line 120 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.4 all_slm

```
real(rprec), dimension(:, :), allocatable vacmod::all_slm
```

Definition at line 139 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.5 all_slp

```
real(rprec), dimension(:, :), allocatable vacmod::all_slp
```

Definition at line 138 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.6 all_tlm

```
real(rprec), dimension(:, :), allocatable vacmod::all_tlm
```

Definition at line 137 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.7 all_tlp

```
real(rprec), dimension(:, :), allocatable vacmod::all_tlp
```

Definition at line 136 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.8 alp

```
real(rprec) vacmod::alp
```

Definition at line 17 of file vacmod.f90.

Referenced by precap().

5.12.2.9 alu

```
real(rprec) vacmod::alu
```

Definition at line 18 of file vacmod.f90.

Referenced by precap().

5.12.2.10 alv

```
real(rprec) vacmod::alv
```

Definition at line 19 of file vacmod.f90.

Referenced by precap().

5.12.2.11 alvp

```
real(rprec) vacmod::alvp
```

Definition at line 20 of file vacmod.f90.

Referenced by precap().

5.12.2.12 amatrix

```
real(rprec), dimension(:), allocatable vacmod::amatrix
```

Definition at line 80 of file vacmod.f90.

Referenced by allocate_nestor(), fouri(), free_mem_nestor(), scalpot(), vacuum(), and nestor_io::write_nestor_↵
outputs().

5.12.2.13 amatsav

```
real(rprec), dimension(:), allocatable vacmod::amatsav
```

Definition at line 32 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `funct3d()`, `nestor()`, `read_nestor_outputs()`, `scalpot()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.14 astemp

```
real(rprec), dimension(:,:,:), allocatable vacmod::astemp
```

Definition at line 157 of file vacmod.f90.

Referenced by `allocate_nestor()`, `fouri()`, and `free_mem_nestor()`.

5.12.2.15 auu

```
real(rprec), dimension(:), allocatable vacmod::auu
```

Definition at line 53 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_↔
outputs()`.

5.12.2.16 auv

```
real(rprec), dimension(:), allocatable vacmod::auv
```

Definition at line 54 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_↔
outputs()`.

5.12.2.17 avv

```
real(rprec), dimension(:), allocatable vacmod::avv
```

Definition at line 55 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_↔
outputs()`.

5.12.2.18 azm1u

```
real(rprec), dimension(:), allocatable vacmod::azm1u
```

Definition at line 131 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.19 azp1u

```
real(rprec), dimension(:), allocatable vacmod::azp1u
```

Definition at line 130 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.20 bcos

```
real(rprec), dimension(:,:,:), allocatable vacmod::bcos
```

Definition at line 152 of file vacmod.f90.

Referenced by `allocate_nestor()`, `fourl()`, and `free_mem_nestor()`.

5.12.2.21 bexn

```
real(rprec), dimension(:), allocatable vacmod::bexn
```

Definition at line 51 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.22 bexni

```
real(rprec), dimension(:), allocatable vacmod::bexni
```

Definition at line 34 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analysum()`, `analysum2()`, `bextern()`, `free_mem_nestor()`, `scalpot()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.23 bexu

```
real(rprec), dimension(:), allocatable vacmod::bexu
```

Definition at line 49 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.24 bexv

```
real(rprec), dimension(:), allocatable vacmod::bexv
```

Definition at line 50 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.25 bphi

```
real(rprec), dimension(:), allocatable vacmod::bphi
```

Definition at line 92 of file vacmod.f90.

Referenced by `allocate_nestor()`, `becoil()`, `bextern()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.26 bphiv

```
real(rprec), dimension(:), allocatable vacmod::bphiv
```

Definition at line 37 of file vacmod.f90.

Referenced by `allocate_nestor()`, `eqfor()`, `free_mem_nestor()`, `freeb_data()`, `read_nestor_outputs()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.27 brad

```
real(rprec), dimension(:), allocatable vacmod::brad
```

Definition at line 91 of file vacmod.f90.

Referenced by `allocate_nestor()`, `becoil()`, `bextern()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.28 brv

```
real(rprec), dimension(:), allocatable vacmod::brv
```

Definition at line 36 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `freeb_data()`, `read_nestor_outputs()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.29 bsin

```
real(rprec), dimension(:,:,:), allocatable vacmod::bsin
```

Definition at line 153 of file vacmod.f90.

Referenced by `allocate_nestor()`, `fouri()`, and `free_mem_nestor()`.

5.12.2.30 bsqvac

```
real(rprec), dimension(:), allocatable vacmod::bsqvac
```

Definition at line 40 of file vacmod.f90.

Referenced by `allocate_nestor()`, `eqfor()`, `free_mem_nestor()`, `freeb_data()`, `funct3d()`, `read_nestor_outputs()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.31 bsubu

```
real(rprec), dimension(:), allocatable vacmod::bsubu
```

Definition at line 76 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.32 bsubv

```
real(rprec), dimension(:), allocatable vacmod::bsubv
```

Definition at line 77 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.33 bsubvvac

```
real(rprec) vacmod::bsubvvac
```

Definition at line 13 of file vacmod.f90.

Referenced by eqfor(), funct3d(), nestor(), read_nestor_outputs(), vacuum(), and nestor_io::write_nestor_outputs().

5.12.2.34 bvecsav

```
real(rprec), dimension(:), allocatable vacmod::bvecsav
```

Definition at line 31 of file vacmod.f90.

Referenced by allocate_nestor(), free_mem_nestor(), funct3d(), nestor(), read_nestor_outputs(), scalpot(), and nestor_io::write_nestor_outputs().

5.12.2.35 bz

```
real(rprec), dimension(:), allocatable vacmod::bz
```

Definition at line 93 of file vacmod.f90.

Referenced by allocate_nestor(), becoil(), belicu(), bextern(), free_mem_nestor(), and nestor_io::write_nestor_outputs().

5.12.2.36 bzv

```
real(rprec), dimension(:), allocatable vacmod::bzv
```

Definition at line 38 of file vacmod.f90.

Referenced by allocate_nestor(), free_mem_nestor(), freeb_data(), read_nestor_outputs(), vacuum(), and nestor_io::write_nestor_outputs().

5.12.2.37 cma

```
real(rprec), dimension(:), allocatable vacmod::cma
```

Definition at line 122 of file vacmod.f90.

Referenced by allocate_nestor(), analyt(), free_mem_nestor(), and nestor_io::write_nestor_outputs().

5.12.2.38 cma11u

```
real(rprec), dimension(:), allocatable vacmod::cma11u
```

Definition at line 132 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.39 delt1u

```
real(rprec), dimension(:), allocatable vacmod::delt1u
```

Definition at line 129 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.40 drv

```
real(rprec), dimension(:), allocatable vacmod::drv
```

Definition at line 61 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.41 dsave

```
real(rprec), dimension(:), allocatable vacmod::dsave
```

Definition at line 145 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, and `greenf()`.

5.12.2.42 g1

```
real(rprec), dimension(:, :, :), allocatable vacmod::g1
```

Definition at line 148 of file vacmod.f90.

Referenced by `allocate_nestor()`, `fourp()`, and `free_mem_nestor()`.

5.12.2.43 g2

```
real(rprec), dimension(:,:), allocatable vacmod::g2
```

Definition at line 149 of file vacmod.f90.

Referenced by `allocate_nestor()`, `fourp()`, and `free_mem_nestor()`.

5.12.2.44 ga1

```
real(rprec), dimension(:), allocatable vacmod::ga1
```

Definition at line 143 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, and `greenf()`.

5.12.2.45 ga2

```
real(rprec), dimension(:), allocatable vacmod::ga2
```

Definition at line 144 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, and `greenf()`.

5.12.2.46 green

```
real(rprec), dimension(:,:), allocatable vacmod::green
```

Definition at line 104 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `scalpot()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.47 greenp

```
real(rprec), dimension(:,:), allocatable vacmod::greenp
```

Definition at line 105 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `scalpot()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.48 grpmn

```
real(rprec), dimension(:), allocatable vacmod::grpmn
```

Definition at line 99 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analysum()`, `analysum2()`, `analyt()`, `fouri()`, `fourp()`, `free_mem_nestor()`, `scalpot()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.49 grpmn_m_map_wrt

```
real(rprec), dimension(:), allocatable vacmod::grpmn_m_map_wrt
```

Definition at line 100 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `scalpot()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.50 grpmn_n_map_wrt

```
real(rprec), dimension(:), allocatable vacmod::grpmn_n_map_wrt
```

Definition at line 101 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `scalpot()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.51 gsave

```
real(rprec), dimension(:), allocatable vacmod::gsave
```

Definition at line 142 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, and `greenf()`.

5.12.2.52 gstore

```
real(rprec), dimension(:), allocatable vacmod::gstore
```

Definition at line 103 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `scalpot()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.53 guu_b

```
real(rprec), dimension(:), allocatable vacmod::guu_b
```

Definition at line 63 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, `greenf()`, `surface()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.54 guv_b

```
real(rprec), dimension(:), allocatable vacmod::guv_b
```

Definition at line 64 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, `greenf()`, `surface()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.55 gvv_b

```
real(rprec), dimension(:), allocatable vacmod::gvv_b
```

Definition at line 65 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, `greenf()`, `surface()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.56 m_map_wrt

```
real(rprec), dimension(:), allocatable vacmod::m_map_wrt
```

Definition at line 28 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.57 n_map_wrt

```
real(rprec), dimension(:), allocatable vacmod::n_map_wrt
```

Definition at line 29 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.58 onp

```
real(rprec) vacmod::onp
```

Definition at line 21 of file vacmod.f90.

Referenced by fouri(), fourp(), precal(), and surface().

5.12.2.59 onp2

```
real(rprec) vacmod::onp2
```

Definition at line 22 of file vacmod.f90.

Referenced by precal(), and surface().

5.12.2.60 p5

```
real(rprec), parameter vacmod::p5 = cp5
```

Definition at line 10 of file vacmod.f90.

Referenced by fouri(), fourp(), precal(), surface(), and vacuum().

5.12.2.61 pi2

```
real(rprec) vacmod::pi2
```

Definition at line 14 of file vacmod.f90.

Referenced by bextern(), fouri(), precal(), and vacuum().

5.12.2.62 pi3

```
real(rprec) vacmod::pi3
```

Definition at line 15 of file vacmod.f90.

Referenced by fouri(), and precal().

5.12.2.63 pi4

```
real(rprec) vacmod::pi4
```

Definition at line 16 of file vacmod.f90.

Referenced by precal().

5.12.2.64 potu

```
real(rprec), dimension(:), allocatable vacmod::potu
```

Definition at line 78 of file vacmod.f90.

Referenced by allocate_nestor(), free_mem_nestor(), vacuum(), and nestor_io::write_nestor_outputs().

5.12.2.65 potv

```
real(rprec), dimension(:), allocatable vacmod::potv
```

Definition at line 79 of file vacmod.f90.

Referenced by allocate_nestor(), free_mem_nestor(), vacuum(), and nestor_io::write_nestor_outputs().

5.12.2.66 potvac

```
real(rprec), dimension(:), allocatable, target vacmod::potvac
```

Definition at line 26 of file vacmod.f90.

Referenced by allocate_nestor(), free_mem_nestor(), freeb_data(), read_nestor_outputs(), vacuum(), nestor_io↵
::write_nestor_outputs(), and wrout().

5.12.2.67 precal_done

```
logical vacmod::precal_done
```

Definition at line 24 of file vacmod.f90.

Referenced by allocate_nestor(), precal(), and vacuum().

5.12.2.68 r0m

```
real(rprec), dimension(:), allocatable vacmod::r0m
```

Definition at line 110 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.69 r0p

```
real(rprec), dimension(:), allocatable vacmod::r0p
```

Definition at line 108 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.70 r1b

```
real(rprec), dimension(:), allocatable vacmod::r1b
```

Definition at line 42 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `greenf()`, `surface()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.71 r1m

```
real(rprec), dimension(:), allocatable vacmod::r1m
```

Definition at line 111 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.72 r1p

```
real(rprec), dimension(:), allocatable vacmod::r1p
```

Definition at line 109 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.73 ra1m

```
real(rprec), dimension(:), allocatable vacmod::ra1m
```

Definition at line 124 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.74 ra1p

```
real(rprec), dimension(:), allocatable vacmod::ra1p
```

Definition at line 123 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.75 raxis_nestor

```
real(rprec), dimension(:), allocatable vacmod::raxis_nestor
```

Definition at line 72 of file vacmod.f90.

Referenced by `allocate_nestor()`, `belicu()`, `free_mem_nestor()`, and `vacuum()`.

5.12.2.76 rcosuv

```
real(rprec), dimension(:), allocatable vacmod::rcosuv
```

Definition at line 69 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.77 rsinuv

```
real(rprec), dimension(:), allocatable vacmod::rsinuv
```

Definition at line 70 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.78 rub

```
real(rprec), dimension(:), allocatable vacmod::rub
```

Definition at line 43 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `surface()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.79 ruu

```
real(rprec), dimension(:), allocatable vacmod::ruu
```

Definition at line 83 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.80 ruv

```
real(rprec), dimension(:), allocatable vacmod::ruv
```

Definition at line 84 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.81 rvb

```
real(rprec), dimension(:), allocatable vacmod::rvb
```

Definition at line 44 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `surface()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.82 rvv

```
real(rprec), dimension(:), allocatable vacmod::rvv
```

Definition at line 85 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.83 rzb2

```
real(rprec), dimension(:), allocatable vacmod::rzb2
```

Definition at line 67 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.84 slm

```
real(rprec), dimension(:), allocatable vacmod::slm
```

Definition at line 125 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analysum2()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.85 slp

```
real(rprec), dimension(:), allocatable vacmod::slp
```

Definition at line 126 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analysum2()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.86 slpm

```
real(rprec), dimension(:), allocatable vacmod::slpm
```

Definition at line 128 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.87 snr

```
real(rprec), dimension(:), allocatable vacmod::snr
```

Definition at line 57 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.88 snv

```
real(rprec), dimension(:), allocatable vacmod::snv
```

Definition at line 58 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_↵`
`outputs()`.

5.12.2.89 snz

```
real(rprec), dimension(:), allocatable vacmod::snz
```

Definition at line 59 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_↵`
`outputs()`.

5.12.2.90 source

```
real(rprec), dimension(:,:,:), allocatable vacmod::source
```

Definition at line 154 of file vacmod.f90.

Referenced by `allocate_nestor()`, `fouri()`, and `free_mem_nestor()`.

5.12.2.91 sqad1u

```
real(rprec), dimension(:), allocatable vacmod::sqad1u
```

Definition at line 133 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.92 sqad2u

```
real(rprec), dimension(:), allocatable vacmod::sqad2u
```

Definition at line 134 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.93 sqрта

```
real(rprec), dimension(:), allocatable vacmod::sqрта
```

Definition at line 113 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.94 sqrtc

```
real(rprec), dimension(:), allocatable vacmod::sqrtc
```

Definition at line 112 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, `free_mem_nestor()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.95 tlm

```
real(rprec), dimension(:), allocatable vacmod::tlm
```

Definition at line 119 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analysum2()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.96 tlm1

```
real(rprec), dimension(:), allocatable vacmod::tlm1
```

Definition at line 118 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.97 tlm2

```
real(rprec), dimension(:), allocatable vacmod::tlm2
```

Definition at line 117 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.98 tlp

```
real(rprec), dimension(:), allocatable vacmod::tlp
```

Definition at line 116 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analysum2()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.99 tlp1

```
real(rprec), dimension(:), allocatable vacmod::tlp1
```

Definition at line 115 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.100 tlp2

```
real(rprec), dimension(:), allocatable vacmod::tlp2
```

Definition at line 114 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.101 tlpm

```
real(rprec), dimension(:), allocatable vacmod::tlpm
```

Definition at line 127 of file vacmod.f90.

Referenced by `allocate_nestor()`, `analyt()`, and `free_mem_nestor()`.

5.12.2.102 two

```
real(rprec), parameter vacmod::two = c2p0
```

Definition at line 11 of file vacmod.f90.

Referenced by `analyt()`.

5.12.2.103 xpts

```
real(rprec), dimension(:,:), allocatable vacmod::xpts
```

Definition at line 96 of file vacmod.f90.

Referenced by `allocate_nestor()`, `belicu()`, and `free_mem_nestor()`.

5.12.2.104 z1b

```
real(rprec), dimension(:), allocatable vacmod::z1b
```

Definition at line 45 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `greenf()`, `surface()`, and `nestor_io::write_nestor_↔
_outputs()`.

5.12.2.105 zaxis_nestor

```
real(rprec), dimension(:), allocatable vacmod::zaxis_nestor
```

Definition at line 73 of file vacmod.f90.

Referenced by `allocate_nestor()`, `belicu()`, `free_mem_nestor()`, and `vacuum()`.

5.12.2.106 zub

```
real(rprec), dimension(:), allocatable vacmod::zub
```

Definition at line 46 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `surface()`, `vacuum()`, and `nestor_io::write_nestor_↔
_outputs()`.

5.12.2.107 zuu

```
real(rprec), dimension(:), allocatable vacmod::zuu
```

Definition at line 86 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.108 zuv

```
real(rprec), dimension(:), allocatable vacmod::zuv
```

Definition at line 87 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.109 zvb

```
real(rprec), dimension(:), allocatable vacmod::zvb
```

Definition at line 47 of file vacmod.f90.

Referenced by `allocate_nestor()`, `bextern()`, `free_mem_nestor()`, `surface()`, `vacuum()`, and `nestor_io::write_nestor_outputs()`.

5.12.2.110 zvv

```
real(rprec), dimension(:), allocatable vacmod::zvv
```

Definition at line 88 of file vacmod.f90.

Referenced by `allocate_nestor()`, `free_mem_nestor()`, `surface()`, and `nestor_io::write_nestor_outputs()`.

5.13 vacmod0 Module Reference**Functions/Subroutines**

- subroutine [set_nestor_sizes](#) (nfp, ntor, mpol, nzeta, ntheta, lasym)

Variables

- integer [mf](#)
- integer [nf](#)
- integer [nu](#)
- integer [nv](#)
- integer [mf1](#)
- integer [nf1](#)
- integer [mnpd](#)
- integer [mnpd2](#)
- integer [nuv](#)
- integer [nu2](#)
- integer [nu3](#)
- integer [nuv2](#)
- integer [nfper](#)
- integer [nvper](#)
- integer [nuv_tan](#)
- integer [nvp](#)
- integer [ndim](#)

5.13.1 Function/Subroutine Documentation

5.13.1.1 set_nestor_sizes()

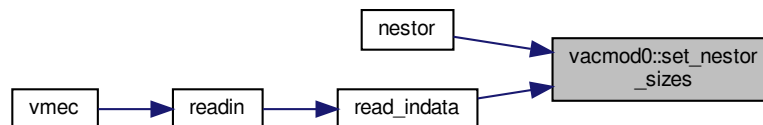
```
subroutine vacmod0::set_nestor_sizes (
    integer, intent(in) nfp,
    integer, intent(in) ntor,
    integer, intent(in) mpol,
    integer, intent(in) nzeta,
    integer, intent(in) ntheta,
    logical, intent(in) lasym )
```

Definition at line 27 of file vacmod0.f90.

References mf, mf1, mnpd, mnpd2, ndim, nf, nf1, nfper, nu, nu2, nu3, nuv, nuv2, nuv_tan, nv, nvp, and nvper.

Referenced by nestor(), and read_indata().

Here is the caller graph for this function:



5.13.2 Variable Documentation

5.13.2.1 mf

```
integer vacmod0::mf
```

Definition at line 5 of file vacmod0.f90.

Referenced by `vacmod0::allocate_nestor()`, and `set_nestor_sizes()`.

5.13.2.2 mf1

```
integer vacmod0::mf1
```

Definition at line 9 of file vacmod0.f90.

Referenced by set_nestor_sizes().

5.13.2.3 mnpd

```
integer vacmod0::mnpd
```

Definition at line 11 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.4 mnpd2

```
integer vacmod0::mnpd2
```

Definition at line 12 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.5 ndim

```
integer vacmod0::ndim
```

Definition at line 23 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.6 nf

```
integer vacmod0::nf
```

Definition at line 6 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.7 nf1

```
integer vacmod0::nf1
```

Definition at line 10 of file vacmod0.f90.

Referenced by set_nestor_sizes().

5.13.2.8 nfper

```
integer vacmod0::nfper
```

Definition at line 17 of file vacmod0.f90.

Referenced by set_nestor_sizes().

5.13.2.9 nu

```
integer vacmod0::nu
```

Definition at line 7 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.10 nu2

```
integer vacmod0::nu2
```

Definition at line 14 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.11 nu3

```
integer vacmod0::nu3
```

Definition at line 15 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.12 nuv

```
integer vacmod0::nuv
```

Definition at line 13 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.13 nuv2

```
integer vacmod0::nuv2
```

Definition at line 16 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.14 nuv_tan

```
integer vacmod0::nuv_tan
```

Definition at line 20 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.15 nv

```
integer vacmod0::nv
```

Definition at line 8 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.16 nvp

```
integer vacmod0::nvp
```

Definition at line 21 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.13.2.17 nvper

```
integer vacmod0::nvper
```

Definition at line 19 of file vacmod0.f90.

Referenced by vacmod::allocate_nestor(), and set_nestor_sizes().

5.14 vforces Module Reference

Variables

- real(rprec), dimension(:), allocatable, target [armn](#)
- real(rprec), dimension(:), allocatable, target [azmn](#)
- real(rprec), dimension(:), allocatable, target [brmn](#)
- real(rprec), dimension(:), allocatable, target [bzm](#)
- real(rprec), dimension(:), allocatable, target [blmn](#)
- real(rprec), dimension(:), allocatable, target [crmn](#)
- real(rprec), dimension(:), allocatable, target [czmn](#)
- real(rprec), dimension(:), allocatable, target [clmn](#)
- real(rprec), dimension(:), pointer [armn_e](#)
- real(rprec), dimension(:), pointer [armn_o](#)
- real(rprec), dimension(:), pointer [azmn_e](#)
- real(rprec), dimension(:), pointer [azmn_o](#)
- real(rprec), dimension(:), pointer [brmn_e](#)
- real(rprec), dimension(:), pointer [brmn_o](#)
- real(rprec), dimension(:), pointer [bzm_e](#)
- real(rprec), dimension(:), pointer [bzm_o](#)
- real(rprec), dimension(:), pointer [blmn_e](#)
- real(rprec), dimension(:), pointer [blmn_o](#)
- real(rprec), dimension(:), pointer [crmn_e](#)
- real(rprec), dimension(:), pointer [crmn_o](#)
- real(rprec), dimension(:), pointer [czmn_e](#)
- real(rprec), dimension(:), pointer [czmn_o](#)
- real(rprec), dimension(:), pointer [clmn_e](#)
- real(rprec), dimension(:), pointer [clmn_o](#)

5.14.1 Variable Documentation

5.14.1.1 armn

```
real(rprec), dimension(:), allocatable, target vforces::armn
```

Definition at line 8 of file vforces.f90.

Referenced by allocate_func3d(), eqfor(), free_mem_func3d(), and func3d().

5.14.1.2 armn_e

```
real(rprec), dimension(:), pointer vforces::armn_e
```

Definition at line 19 of file vforces.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `forces()`, `jacobian()`, and `wrout()`.

5.14.1.3 armn_o

```
real(rprec), dimension(:), pointer vforces::armn_o
```

Definition at line 20 of file vforces.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `eqfor()`, `forces()`, `jacobian()`, and `wrout()`.

5.14.1.4 azmn

```
real(rprec), dimension(:), allocatable, target vforces::azmn
```

Definition at line 9 of file vforces.f90.

Referenced by `allocate_funct3d()`, `eqfor()`, `free_mem_funct3d()`, and `funct3d()`.

5.14.1.5 azmn_e

```
real(rprec), dimension(:), pointer vforces::azmn_e
```

Definition at line 21 of file vforces.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `eqfor()`, `forces()`, `jacobian()`, and `wrout()`.

5.14.1.6 azmn_o

```
real(rprec), dimension(:), pointer vforces::azmn_o
```

Definition at line 22 of file vforces.f90.

Referenced by `allocate_funct3d()`, `bcovar()`, `eqfor()`, `fileout()`, `forces()`, and `jacobian()`.

5.14.1.7 blmn

```
real(rprec), dimension(:), allocatable, target vforces::blmn
```

Definition at line 13 of file vforces.f90.

Referenced by `allocate_func3d()`, `fileout()`, `free_mem_func3d()`, and `func3d()`.

5.14.1.8 blmn_e

```
real(rprec), dimension(:), pointer vforces::blmn_e
```

Definition at line 28 of file vforces.f90.

Referenced by `allocate_func3d()`, and `bcovar()`.

5.14.1.9 blmn_o

```
real(rprec), dimension(:), pointer vforces::blmn_o
```

Definition at line 29 of file vforces.f90.

Referenced by `allocate_func3d()`, and `bcovar()`.

5.14.1.10 brmn

```
real(rprec), dimension(:), allocatable, target vforces::brmn
```

Definition at line 11 of file vforces.f90.

Referenced by `allocate_func3d()`, `eqfor()`, `free_mem_func3d()`, and `func3d()`.

5.14.1.11 brmn_e

```
real(rprec), dimension(:), pointer vforces::brmn_e
```

Definition at line 24 of file vforces.f90.

Referenced by `allocate_func3d()`, `bcovar()`, `eqfor()`, `forces()`, `jacobian()`, and `wrout()`.

5.14.1.12 brmn_o

```
real(rprec), dimension(:), pointer vforces::brmn_o
```

Definition at line 25 of file vforces.f90.

Referenced by `allocate_func3d()`, `bcovar()`, and `forces()`.

5.14.1.13 bzmnn

```
real(rprec), dimension(:), allocatable, target vforces::bzmnn
```

Definition at line 12 of file vforces.f90.

Referenced by `allocate_func3d()`, `eqfor()`, `free_mem_func3d()`, and `func3d()`.

5.14.1.14 bzmnn_e

```
real(rprec), dimension(:), pointer vforces::bzmnn_e
```

Definition at line 26 of file vforces.f90.

Referenced by `allocate_func3d()`, `bcovar()`, `eqfor()`, `forces()`, `jacobian()`, and `wrout()`.

5.14.1.15 bzmnn_o

```
real(rprec), dimension(:), pointer vforces::bzmnn_o
```

Definition at line 27 of file vforces.f90.

Referenced by `allocate_func3d()`, `bcovar()`, `eqfor()`, `fileout()`, `forces()`, and `func3d()`.

5.14.1.16 clmn

```
real(rprec), dimension(:), allocatable, target vforces::clmn
```

Definition at line 17 of file vforces.f90.

Referenced by `allocate_func3d()`, `fileout()`, `free_mem_func3d()`, and `func3d()`.

5.14.1.17 clmn_e

```
real(rprec), dimension(:), pointer vforces::clmn_e
```

Definition at line 35 of file vforces.f90.

Referenced by `allocate_func3d()`, and `bcovar()`.

5.14.1.18 clmn_o

```
real(rprec), dimension(:), pointer vforces::clmn_o
```

Definition at line 36 of file vforces.f90.

Referenced by `allocate_func3d()`, and `bcovar()`.

5.14.1.19 crmn

```
real(rprec), dimension(:), allocatable, target vforces::crmn
```

Definition at line 15 of file vforces.f90.

Referenced by `allocate_func3d()`, `free_mem_func3d()`, and `func3d()`.

5.14.1.20 crmn_e

```
real(rprec), dimension(:), pointer vforces::crmn_e
```

Definition at line 31 of file vforces.f90.

Referenced by `allocate_func3d()`, `eqfor()`, `fileout()`, and `forces()`.

5.14.1.21 crmn_o

```
real(rprec), dimension(:), pointer vforces::crmn_o
```

Definition at line 32 of file vforces.f90.

Referenced by `allocate_func3d()`, `eqfor()`, `fileout()`, and `forces()`.

5.14.1.22 czmn

```
real(rprec), dimension(:), allocatable, target vforces::czmn
```

Definition at line 16 of file vforces.f90.

Referenced by `allocate_func3d()`, `free_mem_func3d()`, and `func3d()`.

5.14.1.23 czmn_e

```
real(rprec), dimension(:), pointer vforces::czmn_e
```

Definition at line 33 of file vforces.f90.

Referenced by `allocate_func3d()`, `eqfor()`, `fileout()`, and `forces()`.

5.14.1.24 czmn_o

```
real(rprec), dimension(:), pointer vforces::czmn_o
```

Definition at line 34 of file vforces.f90.

Referenced by `allocate_func3d()`, `eqfor()`, `forces()`, and `wrout()`.

5.15 vmec_dim Module Reference

Variables

- integer `mpol1`
- integer `ntor1`
- integer `mnmax`
- integer `ntheta1`
- integer `ntheta2`
- integer `ntheta3`
- integer `nznt`
- integer `nrzt`
- integer `mns`
- integer `mnsiz`
- integer `mnmax_nyq`
- integer `ns`
- integer `ns1`
- integer `ns_maxval`

5.15.1 Variable Documentation

5.15.1.1 mnmax

```
integer vmec_dim::mnmax
```

Definition at line 8 of file vmec_dim.f90.

5.15.1.2 mnmax_nyq

```
integer vmec_dim::mnmax_nyq
```

Definition at line 16 of file vmec_dim.f90.

5.15.1.3 mns

```
integer vmec_dim::mns
```

Definition at line 14 of file vmec_dim.f90.

5.15.1.4 mnsiz

```
integer vmec_dim::mnsiz
```

Definition at line 15 of file vmec_dim.f90.

5.15.1.5 mpol1

```
integer vmec_dim::mpol1
```

Definition at line 6 of file vmec_dim.f90.

5.15.1.6 nrzt

```
integer vmec_dim::nrzt
```

Definition at line 13 of file vmec_dim.f90.

Referenced by calc_fbal().

5.15.1.7 ns

```
integer vmec_dim::ns
```

Definition at line 17 of file vmec_dim.f90.

Referenced by calc_fbal(), getcurmid(), jacobian(), and scalfor().

5.15.1.8 ns1

```
integer vmec_dim::ns1
```

Definition at line 18 of file vmec_dim.f90.

Referenced by calc_fbal(), and getcurmid().

5.15.1.9 ns_maxval

```
integer vmec_dim::ns_maxval
```

Definition at line 19 of file vmec_dim.f90.

5.15.1.10 ntheta1

```
integer vmec_dim::ntheta1
```

Definition at line 9 of file vmec_dim.f90.

5.15.1.11 ntheta2

```
integer vmec_dim::ntheta2
```

Definition at line 10 of file vmec_dim.f90.

Referenced by getbsubs(), and getcurmid().

5.15.1.12 ntheta3

```
integer vmec_dim::ntheta3
```

Definition at line 11 of file vmec_dim.f90.

Referenced by getbsubs().

5.15.1.13 ntor1

```
integer vmec_dim::ntor1
```

Definition at line 7 of file vmec_dim.f90.

5.15.1.14 nznt

```
integer vmec_dim::nznt
```

Definition at line 12 of file vmec_dim.f90.

5.16 vmec_input Module Reference**Functions/Subroutines**

- subroutine [read_indata_namelist](#) (iunit, istat)
- subroutine [write_indata_namelist](#) (iunit, istat)

Variables

- integer, parameter [mpol_default](#) = 6
- integer, parameter [ntor_default](#) = 0
- integer, parameter [ns_default](#) = 31
- integer, parameter [niter_default](#) = 100
- real(rprec), parameter [ftol_default](#) = 1.E-10_dp
- integer [nfp](#)
- integer [ncurr](#)
- integer [nstep](#)
- integer [nvacskip](#)
- integer [mpol](#)
- integer [ntor](#)
- integer [ntheta](#)
- integer [nzeta](#)
- integer [mfilter_fbdy](#)
- integer [nfilter_fbdy](#)
- integer, dimension(100) [ns_array](#)

- integer, dimension(100) [niter_array](#)
- real(rprec), dimension(100) [ftol_array](#)
- real(rprec), dimension(-ntord:ntord, 0:mpol1d) [rbc](#)
- real(rprec), dimension(-ntord:ntord, 0:mpol1d) [zbs](#)
- real(rprec), dimension(-ntord:ntord, 0:mpol1d) [rbs](#)
- real(rprec), dimension(-ntord:ntord, 0:mpol1d) [zbc](#)
- real(rprec) [curtor](#)
- real(rprec) [delt](#)
- real(rprec) [tcon0](#)
- real(rprec) [gamma](#)
- real(rprec) [bloat](#)
- real(rprec) [pres_scale](#)
- real(rprec) [spres_ped](#)
 - value of s beyond which pressure profile is flat (pedestal)*
- real(rprec) [phiedge](#)
 - value of real toroidal flux at plasma edge (s=1)*
- real(rprec), dimension(0:20) [am](#)
 - array of coefficients in phi-series for mass (NWT/m**2)*
- real(rprec), dimension(0:20) [ai](#)
 - array of coefficients in phi-series for iota (ncurr=0)*
- real(rprec), dimension(0:20) [ac](#)
 - array of coefficients in phi-series for the quantity $d(l_{\text{curv}})/ds$ = toroidal current density * V_{prime} , so $l_{\text{curv}}(s) = l_{\text{tor}}(s)$ (used for ncurr=1)*
- real(rprec), dimension(1:20) [aphi](#)
- character(len=20) [pcurr_type](#)
- character(len=20) [piota_type](#)
- character(len=20) [pmass_type](#)
- real(rprec), dimension(ndatafmax) [am_aux_s](#)
- real(rprec), dimension(ndatafmax) [am_aux_f](#)
- real(rprec), dimension(ndatafmax) [ai_aux_s](#)
- real(rprec), dimension(ndatafmax) [ai_aux_f](#)
- real(rprec), dimension(ndatafmax) [ac_aux_s](#)
- real(rprec), dimension(ndatafmax) [ac_aux_f](#)
- real(rprec), dimension(0:ntord) [raxis_cc](#)
- real(rprec), dimension(0:ntord) [raxis_cs](#)
- real(rprec), dimension(0:ntord) [zaxis_cc](#)
- real(rprec), dimension(0:ntord) [zaxis_cs](#)
- real(rprec), dimension(nigroup) [extcur](#)
- logical [lfreeb](#)
- logical [lasym](#)
- logical [lbsubs](#)
- character(len=200) [mgrid_file](#)
- character(len=100) [input_extension](#)

5.16.1 Function/Subroutine Documentation

5.16.1.1 read_indata_namelist()

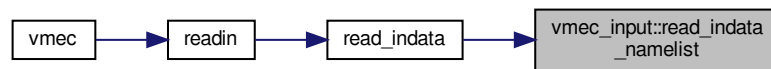
```
subroutine vmec_input::read_indata_namelist (
    integer iunit,
    integer istat )
```

Definition at line 87 of file vmec_input.f90.

References ac, ac_aux_s, ai, ai_aux_s, am, am_aux_s, aphi, bloat, curtor, delt, extcur, ftol_array, ftol_default, gamma, lasym, lbsubs, lfreeb, mfilter_fbdy, mgrid_file, mpol, mpol_default, ncurr, nfilter_fbdy, nfp, niter_array, niter_↵_default, ns_array, ns_default, nstep, ntheta, ntor, ntor_default, nvacskip, nzeta, pcurr_type, phiedge, piota_type, pmass_type, pres_scale, raxis_cc, raxis_cs, rbc, rbs, spres_ped, tcon0, zaxis_cc, zaxis_cs, zbc, and zbs.

Referenced by read_indata().

Here is the caller graph for this function:



5.16.1.2 write_indata_namelist()

```
subroutine vmec_input::write_indata_namelist (
    integer, intent(in) iunit,
    integer, intent(inout) istat )
```

Definition at line 133 of file vmec_input.f90.

References ac, ac_aux_f, ac_aux_s, ai, ai_aux_f, ai_aux_s, am, am_aux_f, am_aux_s, bloat, curtor, delt, extcur, ftol_array, gamma, lasym, lfreeb, mgrid_file, mpol, ncurr, nfp, niter_array, ns_array, nstep, ntor, nvacskip, nzeta, pcurr_type, phiedge, piota_type, pmass_type, pres_scale, raxis_cc, raxis_cs, rbc, rbs, spres_ped, tcon0, zaxis_cc, zaxis_cs, zbc, and zbs.

5.16.2 Variable Documentation

5.16.2.1 ac

```
real(rprec), dimension(0:20) vmec_input::ac
```

array of coefficients in phi-series for the quantity $d(l_{\text{curv}})/ds$ = toroidal current density * V_{prime} , so $l_{\text{curv}}(s) = l_{\text{tor}}(s)$ (used for $ncurr=1$)

Definition at line 44 of file vmec_input.f90.

Referenced by pcurr(), read_indata_namelist(), and write_indata_namelist().

5.16.2.2 ac_aux_f

```
real(rprec), dimension(ndatafmax) vmec_input::ac_aux_f
```

Definition at line 55 of file vmec_input.f90.

Referenced by pcurr(), and write_indata_namelist().

5.16.2.3 ac_aux_s

```
real(rprec), dimension(ndatafmax) vmec_input::ac_aux_s
```

Definition at line 54 of file vmec_input.f90.

Referenced by pcurr(), read_indata_namelist(), and write_indata_namelist().

5.16.2.4 ai

```
real(rprec), dimension(0:20) vmec_input::ai
```

array of coefficients in phi-series for iota (ncurr=0)

Definition at line 43 of file vmec_input.f90.

Referenced by piota(), read_indata_namelist(), and write_indata_namelist().

5.16.2.5 ai_aux_f

```
real(rprec), dimension(ndatafmax) vmec_input::ai_aux_f
```

Definition at line 53 of file vmec_input.f90.

Referenced by piota(), and write_indata_namelist().

5.16.2.6 ai_aux_s

```
real(rprec), dimension(ndatafmax) vmec_input::ai_aux_s
```

Definition at line 52 of file vmec_input.f90.

Referenced by piota(), read_indata_namelist(), and write_indata_namelist().

5.16.2.7 am

```
real(rprec), dimension(0:20) vmec_input::am
```

array of coefficients in phi-series for mass (NWT/m**2)

Definition at line 42 of file vmec_input.f90.

Referenced by pmass(), read_indata_namelist(), and write_indata_namelist().

5.16.2.8 am_aux_f

```
real(rprec), dimension(ndatafmax) vmec_input::am_aux_f
```

Definition at line 51 of file vmec_input.f90.

Referenced by pmass(), and write_indata_namelist().

5.16.2.9 am_aux_s

```
real(rprec), dimension(ndatafmax) vmec_input::am_aux_s
```

Definition at line 50 of file vmec_input.f90.

Referenced by pmass(), read_indata_namelist(), and write_indata_namelist().

5.16.2.10 aphi

```
real(rprec), dimension(1:20) vmec_input::aphi
```

Definition at line 46 of file vmec_input.f90.

Referenced by read_indata_namelist(), and torflux_deriv().

5.16.2.11 bloat

```
real(rprec) vmec_input::bloat
```

Definition at line 38 of file vmec_input.f90.

Referenced by pcurr(), pmass(), read_indata(), read_indata_namelist(), and write_indata_namelist().

5.16.2.12 **curtor**

```
real(rprec) vmec_input::curtor
```

Definition at line 34 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.13 **delt**

```
real(rprec) vmec_input::delt
```

Definition at line 35 of file vmec_input.f90.

Referenced by read_indata_namelist(), reset_params(), vmec(), and write_indata_namelist().

5.16.2.14 **extcur**

```
real(rprec), dimension(nigroup) vmec_input::extcur
```

Definition at line 61 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.15 **ftol_array**

```
real(rprec), dimension(100) vmec_input::ftol_array
```

Definition at line 28 of file vmec_input.f90.

Referenced by read_indata_namelist(), vmec(), write_indata_namelist(), and wrout().

5.16.2.16 **ftol_default**

```
real(rprec), parameter vmec_input::ftol_default = 1.E-10_dp
```

Definition at line 13 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.17 gamma

```
real(rprec) vmec_input::gamma
```

Definition at line 37 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.18 input_extension

```
character(len=100) vmec_input::input_extension
```

Definition at line 74 of file vmec_input.f90.

Referenced by mercier(), and vmec().

5.16.2.19 lasym

```
logical vmec_input::lasym
```

Definition at line 63 of file vmec_input.f90.

Referenced by getbsubs(), read_indata_namelist(), and write_indata_namelist().

5.16.2.20 lbsubs

```
logical vmec_input::lbsubs
```

Definition at line 71 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.21 lfreeb

```
logical vmec_input::lfreeb
```

Definition at line 62 of file vmec_input.f90.

Referenced by read_indata_namelist(), mgrid_mod::read_mgrid(), vmec(), and write_indata_namelist().

5.16.2.22 mfilter_fbdy

```
integer vmec_input::mfilter_fbdy
```

Definition at line 23 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.23 mgrid_file

```
character(len=200) vmec_input::mgrid_file
```

Definition at line 73 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.24 mpol

```
integer vmec_input::mpol
```

Definition at line 19 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.25 mpol_default

```
integer, parameter vmec_input::mpol_default = 6
```

Definition at line 9 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.26 ncurr

```
integer vmec_input::ncurr
```

Definition at line 16 of file vmec_input.f90.

Referenced by read_indata(), read_indata_namelist(), and write_indata_namelist().

5.16.2.27 nfilter_fbdy

```
integer vmec_input::nfilter_fbdy
```

Definition at line 24 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.28 nfp

```
integer vmec_input::nfp
```

Definition at line 15 of file vmec_input.f90.

Referenced by getbsubs(), read_indata_namelist(), and write_indata_namelist().

5.16.2.29 niter_array

```
integer, dimension(100) vmec_input::niter_array
```

Definition at line 27 of file vmec_input.f90.

Referenced by read_indata_namelist(), vmec(), and write_indata_namelist().

5.16.2.30 niter_default

```
integer, parameter vmec_input::niter_default = 100
```

Definition at line 12 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.31 ns_array

```
integer, dimension(100) vmec_input::ns_array
```

Definition at line 26 of file vmec_input.f90.

Referenced by read_indata_namelist(), vmec(), write_indata_namelist(), and wrout().

5.16.2.32 ns_default

```
integer, parameter vmec_input::ns_default = 31
```

Definition at line 11 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.33 nstep

```
integer vmec_input::nstep
```

Definition at line 17 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.34 ntheta

```
integer vmec_input::ntheta
```

Definition at line 21 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.35 ntor

```
integer vmec_input::ntor
```

Definition at line 20 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.36 ntor_default

```
integer, parameter vmec_input::ntor_default = 0
```

Definition at line 10 of file vmec_input.f90.

Referenced by read_indata_namelist().

5.16.2.37 nvacskip

```
integer vmec_input::nvacskip
```

Definition at line 18 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.38 nzeta

```
integer vmec_input::nzeta
```

Definition at line 22 of file vmec_input.f90.

Referenced by getbsubs(), getcurmid(), read_indata_namelist(), and write_indata_namelist().

5.16.2.39 pcurr_type

```
character(len=20) vmec_input::pcurr_type
```

Definition at line 47 of file vmec_input.f90.

Referenced by pcurr(), read_indata_namelist(), and write_indata_namelist().

5.16.2.40 phiedge

```
real(rprec) vmec_input::phiedge
```

value of real toroidal flux at plasma edge (s=1)

Definition at line 41 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.41 piota_type

```
character(len=20) vmec_input::piota_type
```

Definition at line 48 of file vmec_input.f90.

Referenced by piota(), read_indata_namelist(), and write_indata_namelist().

5.16.2.42 pmass_type

```
character(len=20) vmec_input::pmass_type
```

Definition at line 49 of file vmec_input.f90.

Referenced by pmass(), read_indata_namelist(), and write_indata_namelist().

5.16.2.43 pres_scale

```
real(rprec) vmec_input::pres_scale
```

Definition at line 39 of file vmec_input.f90.

Referenced by pmass(), read_indata_namelist(), and write_indata_namelist().

5.16.2.44 raxis_cc

```
real(rprec), dimension(0:ntord) vmec_input::raxis_cc
```

Definition at line 57 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.45 raxis_cs

```
real(rprec), dimension(0:ntord) vmec_input::raxis_cs
```

Definition at line 58 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.46 rbc

```
real(rprec), dimension(-ntord:ntord,0:mpolld) vmec_input::rbc
```

Definition at line 30 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.47 rbs

```
real(rprec), dimension(-ntord:ntord,0:mpolld) vmec_input::rbs
```

Definition at line 32 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.48 spres_ped

```
real(rprec) vmec_input::spres_ped
```

value of s beyond which pressure profile is flat (pedestal)

Definition at line 40 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.49 tcon0

```
real(rprec) vmec_input::tcon0
```

Definition at line 36 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.50 zaxis_cc

```
real(rprec), dimension(0:ntord) vmec_input::zaxis_cc
```

Definition at line 59 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.51 zaxis_cs

```
real(rprec), dimension(0:ntord) vmec_input::zaxis_cs
```

Definition at line 60 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.52 zbc

```
real(rprec), dimension(-ntord:ntord,0:mpolld) vmec_input::zbc
```

Definition at line 33 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.16.2.53 zbs

```
real(rprec), dimension(-ntord:ntord,0:mpolld) vmec_input::zbs
```

Definition at line 31 of file vmec_input.f90.

Referenced by read_indata_namelist(), and write_indata_namelist().

5.17 vmec_io Module Reference

Variables

- real(rprec) [volavgb](#)
- real(rprec) [ionlarmor](#)
- real(rprec) [aminor_p](#)
- real(rprec) [rmajor_p](#)
- real(rprec) [betatot](#)
- real(rprec) [betapol](#)
- real(rprec) [betator](#)
- real(rprec) [betaxis](#)
- real(rprec) [b0](#)
- real(rprec) [volume_p](#)
- real(rprec) [cross_area_p](#)
- real(rprec) [surf_area_p](#)
- real(rprec) [circum_p](#)
- real(rprec) [kappa_p](#)
- real(rprec) [rmax_surf](#)
- real(rprec) [rmin_surf](#)
- real(rprec) [zmax_surf](#)

5.17.1 Variable Documentation

5.17.1.1 aminor_p

```
real(rprec) vmec_io::aminor_p
```

Definition at line 10 of file vmec_io.f90.

Referenced by aspectratio(), eqfor(), and wrout().

5.17.1.2 b0

```
real(rprec) vmec_io::b0
```

Definition at line 16 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.3 betapol

```
real(rprec) vmec_io::betapol
```

Definition at line 13 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.4 betator

```
real(rprec) vmec_io::betator
```

Definition at line 14 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.5 betatot

```
real(rprec) vmec_io::betatot
```

Definition at line 12 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.6 betaxis

```
real(rprec) vmec_io::betaxis
```

Definition at line 15 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.7 **circum_p**

```
real(rprec) vmec_io::circum_p
```

Definition at line 20 of file vmec_io.f90.

Referenced by eqfor().

5.17.1.8 **cross_area_p**

```
real(rprec) vmec_io::cross_area_p
```

Definition at line 18 of file vmec_io.f90.

Referenced by aspectratio(), and eqfor().

5.17.1.9 **ionlarmor**

```
real(rprec) vmec_io::ionlarmor
```

Definition at line 9 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.10 **kappa_p**

```
real(rprec) vmec_io::kappa_p
```

Definition at line 21 of file vmec_io.f90.

Referenced by eqfor().

5.17.1.11 **rmajor_p**

```
real(rprec) vmec_io::rmajor_p
```

Definition at line 11 of file vmec_io.f90.

Referenced by aspectratio(), eqfor(), and wrout().

5.17.1.12 rmax_surf

```
real(rprec) vmec_io::rmax_surf
```

Definition at line 22 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.13 rmin_surf

```
real(rprec) vmec_io::rmin_surf
```

Definition at line 23 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.14 surf_area_p

```
real(rprec) vmec_io::surf_area_p
```

Definition at line 19 of file vmec_io.f90.

Referenced by eqfor().

5.17.1.15 volavgb

```
real(rprec) vmec_io::volavgb
```

Definition at line 8 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.17.1.16 volume_p

```
real(rprec) vmec_io::volume_p
```

Definition at line 17 of file vmec_io.f90.

Referenced by aspectratio(), eqfor(), and wrout().

5.17.1.17 zmax_surf

```
real(rprec) vmec_io::zmax_surf
```

Definition at line 24 of file vmec_io.f90.

Referenced by eqfor(), and wrout().

5.18 vmec_main Module Reference

Variables

- real(rprec), dimension(:,:), allocatable [ard](#)
- real(rprec), dimension(:,:), allocatable [arm](#)
- real(rprec), dimension(:,:), allocatable [brd](#)
- real(rprec), dimension(:,:), allocatable [brm](#)
- real(rprec), dimension(:,:), allocatable [azd](#)
- real(rprec), dimension(:,:), allocatable [azm](#)
- real(rprec), dimension(:,:), allocatable [bzd](#)
- real(rprec), dimension(:,:), allocatable [bzm](#)
- real(rprec), dimension(:,:), allocatable [bmin](#)
- real(rprec), dimension(:,:), allocatable [bmax](#)
- real(rprec), dimension(:), allocatable [crd](#)
- real(rprec), dimension(:), allocatable [iotaf](#)
- real(rprec), dimension(:), allocatable [hipf](#)
- real(rprec), dimension(:), allocatable [chipf](#)
- real(rprec), dimension(:), allocatable [phi](#)
- real(rprec), dimension(:), allocatable [beta_vol](#)
- real(rprec), dimension(:), allocatable [jcuru](#)
- real(rprec), dimension(:), allocatable [jcurv](#)
- real(rprec), dimension(:), allocatable [jdotb](#)
- real(rprec), dimension(:), allocatable [buco](#)
- real(rprec), dimension(:), allocatable [bvco](#)
- real(rprec), dimension(:), allocatable [bdotgradv](#)
- real(rprec), dimension(:), allocatable [equip](#)
- real(rprec), dimension(:), allocatable [specw](#)
- real(rprec), dimension(:), allocatable [tcon](#)
- real(rprec), dimension(:), allocatable [psi](#)
- real(rprec), dimension(:), allocatable [yellip](#)
- real(rprec), dimension(:), allocatable [yinden](#)
- real(rprec), dimension(:), allocatable [ytrian](#)
- real(rprec), dimension(:), allocatable [yshift](#)
- real(rprec), dimension(:), allocatable [ygeo](#)
- real(rprec), dimension(:), allocatable [overr](#)
- real(rprec), dimension(:), allocatable [sm](#)
- real(rprec), dimension(:), allocatable [sp](#)
- real(rprec), dimension(:), allocatable [pres](#)
- real(rprec), dimension(:), allocatable [vp](#)
- real(rprec), dimension(:), allocatable [jpar2](#)
- real(rprec), dimension(:), allocatable [jperp2](#)
- real(rprec), dimension(:), allocatable [bdotb](#)
- real(rprec), dimension(:), allocatable [blam](#)

- real(rprec), dimension(:), allocatable [clam](#)
- real(rprec), dimension(:), allocatable [dlam](#)
- real(rprec), dimension(:), allocatable [vpphi](#)
- real(rprec), dimension(:), allocatable [presgrad](#)
- real(rprec), dimension(:), allocatable [bdamp](#)
- real(rprec), dimension(:), allocatable [bucof](#)
- real(rprec), dimension(:), allocatable [bvcof](#)
- real(rprec), dimension(:), allocatable [chi](#)
- real(rprec), dimension(:), allocatable [presf](#)
*pressure profile on full-grid, mass/phi**gamma*
- real(rprec), dimension(:), allocatable [chips](#)
poloidal flux (same as chip), one-dimensional array
- real(rprec), dimension(:), allocatable [phips](#)
toroidal flux (same as phi), one-dimensional array
- real(rprec), dimension(:), allocatable [iotas](#)
rotational transform , on half radial mesh
- real(rprec), dimension(:), allocatable [icurv](#)
(-)toroidal current inside flux surface (vanishes like s)
- real(rprec), dimension(:), allocatable [mass](#)
mass profile on half-grid
- real(rprec), dimension(:,:,:), allocatable [faclam](#)
- real(rprec), dimension(:,:,:), allocatable [faclam0](#)
- real(rprec), dimension(:,:), allocatable [bsqsav](#)
- real(rprec), dimension(:), allocatable [bledge](#)
- real(rprec), dimension(:), allocatable [bpedge](#)
- real(rprec), dimension(:), allocatable [bzedge](#)
- real(rprec), dimension(:), allocatable [xcl0](#)
- real(rprec), dimension(0:mpol1d, 3) [xmpq](#)
- real(rprec), dimension(0:mpol1d) [faccon](#)
- real(rprec) [hs](#)
radial mesh size increment
- real(rprec) [currv](#)
- real(rprec) [aspect](#)
- real(rprec) [ohs](#)
- real(rprec) [voli](#)
- real(rprec) [r00](#)
- real(rprec) [r0scale](#)
- real(rprec) [z00](#)
- real(rprec) [fsqsum0](#)
- real(rprec) [fnorm](#)
- real(rprec) [fsqr](#) =1
- real(rprec) [fsqz](#) =1
- real(rprec) [fsql](#) =1
- real(rprec) [fnorm1](#)
- real(rprec) [fnorml](#)
- real(rprec) [fsqr1](#)
- real(rprec) [fsqz1](#)
- real(rprec) [fsql1](#)
- real(rprec) [fsq](#)
- real(rprec) [fedge](#)
- real(rprec) [wb](#)
- real(rprec) [wp](#)
- real(rprec) [router](#)

- real(rprec) [rinner](#)
- real(rprec) [ftolv](#)
- real(rprec) [otav](#)
- time-step algorithm*
- real(rprec), dimension(ndamp) [otau](#)
- real(rprec), dimension(:, :, :), allocatable, target [rmn_bdy](#)
- real(rprec), dimension(:, :, :), allocatable, target [zmn_bdy](#)
- real(rprec), dimension(:), allocatable [bsubu0](#)
- real(rprec), dimension(:), allocatable [dbsq](#)
- real(rprec), dimension(:), allocatable [rbsq](#)
- real(rprec) [rbtor](#)
- real(rprec) [rbtor0](#)
- real(rprec) [ctor](#)
- real(rprec) [delbsq](#)
- real(rprec) [res0](#)
- real(rprec) [delt0r](#)
- real(rprec), dimension(ndatafmax) [spfa](#)
- real(rprec), dimension(ndatafmax) [spfa2](#)
- real(rprec), dimension(ndatafmax) [hp](#)
- real(rprec), dimension(ndatafmax) [sifa](#)
- real(rprec), dimension(ndatafmax) [sifa2](#)
- real(rprec), dimension(ndatafmax) [hi](#)
- logical [lthreed](#)
- logical [lconm1](#)
- logical [lflip](#)
- from init_geometry*
- integer, dimension(:), allocatable [ireflect](#)
- two-dimensional array for computing 2pi-v angle*
- integer [multi_ns_grid](#)
- integer [itfsq](#)
- integer [ndatap](#)
- integer [ndatai](#)
- integer [niter](#)
- max iterations for current multi-grid iteration*
- integer [neqs](#)
- total number of equations to evolve (size of xc)*
- integer [irzloff](#)
- offset in xc array between R,Z,L components*
- integer [iequi](#)
- counter used to call -EQFOR- at end of run*
- integer [ijacob](#)
- counter for number of times jacobian changes sign*
- integer [irst](#)
- "counter" monitoring sign of jacobian; resets R, Z, and Lambda when jacobian changes sign and decreases time step*
- integer [iter1](#)
- number of iterations at which the currently active evolution was branched off from*
- integer [iter2](#)
- total number of iterations*
- integer [ivac](#)
- counts number of free-boundary iterations*
- integer [vacuum_calls](#) = 0

5.18.1 Variable Documentation

5.18.1.1 ard

```
real(rprec), dimension(:,:), allocatable vmec_main::ard
```

Definition at line 12 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, `residue()`, and `scale_m1()`.

5.18.1.2 arm

```
real(rprec), dimension(:,:), allocatable vmec_main::arm
```

Definition at line 13 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, and `residue()`.

5.18.1.3 aspect

```
real(rprec) vmec_main::aspect
```

Definition at line 85 of file vmec_main.f90.

Referenced by `eqfor()`, and `wrout()`.

5.18.1.4 azd

```
real(rprec), dimension(:,:), allocatable vmec_main::azd
```

Definition at line 16 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, `residue()`, and `scale_m1()`.

5.18.1.5 azm

```
real(rprec), dimension(:,:), allocatable vmec_main::azm
```

Definition at line 17 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, and `residue()`.

5.18.1.6 bdamp

```
real(rprec), dimension(:), allocatable vmec_main::bdamp
```

Definition at line 57 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, and `profil1d()`.

5.18.1.7 bdotb

```
real(rprec), dimension(:), allocatable vmec_main::bdotb
```

Definition at line 51 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, `free_mem_ns()`, and `jxbforce()`.

5.18.1.8 bdotgradv

```
real(rprec), dimension(:), allocatable vmec_main::bdotgradv
```

Definition at line 34 of file vmec_main.f90.

Referenced by `allocate_ns()`, `free_mem_ns()`, `jxbforce()`, and `wrout()`.

5.18.1.9 beta_vol

```
real(rprec), dimension(:), allocatable vmec_main::beta_vol
```

Definition at line 28 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, `free_mem_ns()`, and `wrout()`.

5.18.1.10 blam

```
real(rprec), dimension(:), allocatable vmec_main::blam
```

Definition at line 52 of file vmec_main.f90.

Referenced by `allocate_ns()`, `free_mem_ns()`, and `lamcal()`.

5.18.1.11 bmax

```
real(rprec), dimension(:,:), allocatable vmec_main::bmax
```

Definition at line 21 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, and `free_mem_ns()`.

5.18.1.12 bmin

```
real(rprec), dimension(:,:), allocatable vmec_main::bmin
```

Definition at line 20 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, and `free_mem_ns()`.

5.18.1.13 bpedge

```
real(rprec), dimension(:), allocatable vmec_main::bpedge
```

Definition at line 75 of file vmec_main.f90.

Referenced by `eqfor()`, and `freeb_data()`.

5.18.1.14 brd

```
real(rprec), dimension(:,:), allocatable vmec_main::brd
```

Definition at line 14 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, `residue()`, and `scale_m1()`.

5.18.1.15 bredge

```
real(rprec), dimension(:), allocatable vmec_main::bredge
```

Definition at line 74 of file vmec_main.f90.

Referenced by `eqfor()`, and `freeb_data()`.

5.18.1.16 brm

```
real(rprec), dimension(:,:), allocatable vmec_main::brm
```

Definition at line 15 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, and `residue()`.

5.18.1.17 bsqsav

```
real(rprec), dimension(:,:), allocatable vmec_main::bsqsav
```

Definition at line 72 of file vmec_main.f90.

Referenced by `allocate_nunv()`, `free_mem_nunv()`, `freeb_data()`, `funct3d()`, and `printout()`.

5.18.1.18 bsubu0

```
real(rprec), dimension(:), allocatable vmec_main::bsubu0
```

Definition at line 118 of file vmec_main.f90.

Referenced by `allocate_nunv()`, and `free_mem_nunv()`.

5.18.1.19 buco

```
real(rprec), dimension(:), allocatable vmec_main::buco
```

Definition at line 32 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `calc_fbal()`, `eqfor()`, `free_mem_ns()`, and `wrout()`.

5.18.1.20 bucof

```
real(rprec), dimension(:), allocatable vmec_main::bucof
```

Definition at line 58 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, and `free_mem_ns()`.

5.18.1.21 bvco

```
real(rprec), dimension(:), allocatable vmec_main::bvco
```

Definition at line 33 of file vmec_main.f90.

Referenced by allocate_ns(), bcovar(), calc_fbal(), eqfor(), free_mem_ns(), and wrout().

5.18.1.22 bvcof

```
real(rprec), dimension(:), allocatable vmec_main::bvcof
```

Definition at line 59 of file vmec_main.f90.

Referenced by allocate_ns(), eqfor(), and free_mem_ns().

5.18.1.23 bzd

```
real(rprec), dimension(:, :), allocatable vmec_main::bzd
```

Definition at line 18 of file vmec_main.f90.

Referenced by allocate_ns(), bcovar(), free_mem_ns(), residue(), and scale_m1().

5.18.1.24 bzedge

```
real(rprec), dimension(:), allocatable vmec_main::bzedge
```

Definition at line 76 of file vmec_main.f90.

Referenced by eqfor(), and freeb_data().

5.18.1.25 bzm

```
real(rprec), dimension(:, :), allocatable vmec_main::bzm
```

Definition at line 19 of file vmec_main.f90.

Referenced by allocate_ns(), bcovar(), free_mem_ns(), and residue().

5.18.1.26 chi

```
real(rprec), dimension(:), allocatable vmec_main::chi
```

Definition at line 60 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, `free_mem_ns()`, and `wrout()`.

5.18.1.27 chipf

```
real(rprec), dimension(:), allocatable vmec_main::chipf
```

Definition at line 26 of file vmec_main.f90.

Referenced by `add_fluxes()`, `allocate_ns()`, `calc_fbal()`, `eqfor()`, `free_mem_ns()`, `profil1d()`, and `wrout()`.

5.18.1.28 chips

```
real(rprec), dimension(:), allocatable vmec_main::chips
```

poloidal flux (same as chip), one-dimensional array

Definition at line 63 of file vmec_main.f90.

Referenced by `add_fluxes()`, `allocate_ns()`, `free_mem_ns()`, `profil1d()`, and `profil3d()`.

5.18.1.29 clam

```
real(rprec), dimension(:), allocatable vmec_main::clam
```

Definition at line 53 of file vmec_main.f90.

Referenced by `allocate_ns()`, `free_mem_ns()`, and `lamcal()`.

5.18.1.30 crd

```
real(rprec), dimension(:), allocatable vmec_main::crd
```

Definition at line 23 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, and `residue()`.

5.18.1.31 ctor

```
real(rprec) vmec_main::ctor
```

Definition at line 124 of file vmec_main.f90.

Referenced by bcovar(), eqfor(), funct3d(), and wrout().

5.18.1.32 currv

```
real(rprec) vmec_main::currv
```

Definition at line 84 of file vmec_main.f90.

Referenced by profil1d(), and readin().

5.18.1.33 dbsq

```
real(rprec), dimension(:), allocatable vmec_main::dbsq
```

Definition at line 119 of file vmec_main.f90.

Referenced by allocate_nunv(), free_mem_nunv(), funct3d(), and printout().

5.18.1.34 delbsq

```
real(rprec) vmec_main::delbsq
```

Definition at line 125 of file vmec_main.f90.

Referenced by printout().

5.18.1.35 delt0r

```
real(rprec) vmec_main::delt0r
```

Definition at line 127 of file vmec_main.f90.

Referenced by eqsolve(), reset_params(), and vmec().

5.18.1.36 dlam

```
real(rprec), dimension(:), allocatable vmec_main::dlam
```

Definition at line 54 of file vmec_main.f90.

Referenced by `allocate_ns()`, `free_mem_ns()`, and `lamcal()`.

5.18.1.37 equip

```
real(rprec), dimension(:), allocatable vmec_main::equip
```

Definition at line 35 of file vmec_main.f90.

Referenced by `allocate_ns()`, `calc_fbal()`, `eqfor()`, `free_mem_ns()`, and `wrout()`.

5.18.1.38 faccon

```
real(rprec), dimension(0:mpolld) vmec_main::faccon
```

Definition at line 81 of file vmec_main.f90.

Referenced by `alias()`, and `fixaray()`.

5.18.1.39 faclam

```
real(rprec), dimension(:,:,:), allocatable vmec_main::faclam
```

Definition at line 69 of file vmec_main.f90.

Referenced by `allocate_ns()`, `free_mem_ns()`, `lamcal()`, `profil3d()`, and `residue()`.

5.18.1.40 faclam0

```
real(rprec), dimension(:,:,:), allocatable vmec_main::faclam0
```

Definition at line 70 of file vmec_main.f90.

5.18.1.41 fedge

```
real(rprec) vmec_main::fedge
```

Definition at line 102 of file vmec_main.f90.

Referenced by printout(), and residue().

5.18.1.42 fnorm

```
real(rprec) vmec_main::fnorm
```

Definition at line 92 of file vmec_main.f90.

Referenced by bcovar(), and residue().

5.18.1.43 fnorm1

```
real(rprec) vmec_main::fnorm1
```

Definition at line 96 of file vmec_main.f90.

Referenced by bcovar(), and residue().

5.18.1.44 fnorml

```
real(rprec) vmec_main::fnorml
```

Definition at line 97 of file vmec_main.f90.

Referenced by bcovar(), and residue().

5.18.1.45 fsq

```
real(rprec) vmec_main::fsq
```

Definition at line 101 of file vmec_main.f90.

Referenced by eqsolve(), evolve(), initialize_radial(), and reset_params().

5.18.1.46 fsql

```
real(rprec) vmec_main::fsql =1
```

Definition at line 95 of file vmec_main.f90.

Referenced by evolve(), funct3d(), printout(), residue(), and wrout().

5.18.1.47 fsql1

```
real(rprec) vmec_main::fsql1
```

Definition at line 100 of file vmec_main.f90.

Referenced by evolve(), printout(), and residue().

5.18.1.48 fsqr

```
real(rprec) vmec_main::fsqr =1
```

Definition at line 93 of file vmec_main.f90.

Referenced by eqsolve(), evolve(), funct3d(), printout(), reset_params(), residue(), and wrout().

5.18.1.49 fsqr1

```
real(rprec) vmec_main::fsqr1
```

Definition at line 98 of file vmec_main.f90.

Referenced by evolve(), printout(), and residue().

5.18.1.50 fsqsum0

```
real(rprec) vmec_main::fsqsum0
```

Definition at line 91 of file vmec_main.f90.

5.18.1.51 fsqz

```
real(rprec) vmec_main::fsqz =1
```

Definition at line 94 of file vmec_main.f90.

Referenced by constrain_m1(), eqsolve(), evolve(), funct3d(), printout(), reset_params(), residue(), and wrout().

5.18.1.52 fsqz1

```
real(rprec) vmec_main::fsqz1
```

Definition at line 99 of file vmec_main.f90.

Referenced by evolve(), printout(), and residue().

5.18.1.53 ftolv

```
real(rprec) vmec_main::ftolv
```

Definition at line 109 of file vmec_main.f90.

Referenced by evolve(), initialize_radial(), reset_params(), and vmec().

5.18.1.54 hi

```
real(rprec), dimension(ndatafmax) vmec_main::hi
```

Definition at line 134 of file vmec_main.f90.

5.18.1.55 hp

```
real(rprec), dimension(ndatafmax) vmec_main::hp
```

Definition at line 131 of file vmec_main.f90.

5.18.1.56 hs

```
real(rprec) vmec_main::hs
```

radial mesh size increment

Definition at line 83 of file vmec_main.f90.

Referenced by bcovar(), eqfor(), fileout(), funct3d(), guess_axis(), initialize_radial(), preconfn(), profil1d(), residue(), and scalfor().

5.18.1.57 icurv

```
real(rprec), dimension(:), allocatable vmec_main::icurv
```

(-)toroidal current inside flux surface (vanishes like s)

Definition at line 66 of file vmec_main.f90.

Referenced by add_fluxes(), allocate_ns(), free_mem_ns(), and profil1d().

5.18.1.58 iequi

```
integer vmec_main::iequi
```

counter used to call -EQFOR- at end of run

Definition at line 150 of file vmec_main.f90.

Referenced by bcovar(), fileout(), funct3d(), and vmec().

5.18.1.59 ijacob

```
integer vmec_main::ijacob
```

counter for number of times jacobian changes sign

Definition at line 151 of file vmec_main.f90.

Referenced by eqsolve(), fileout(), initialize_radial(), reset_params(), and restart_iter().

5.18.1.60 iotaf

```
real(rprec), dimension(:), allocatable vmec_main::iotaf
```

Definition at line 24 of file vmec_main.f90.

Referenced by add_fluxes(), allocate_ns(), eqfor(), free_mem_ns(), profil1d(), and wrout().

5.18.1.61 iotas

```
real(rprec), dimension(:), allocatable vmec_main::iotas
```

rotational transform , on half radial mesh

Definition at line 65 of file vmec_main.f90.

Referenced by add_fluxes(), allocate_ns(), eqfor(), free_mem_ns(), jxbforce(), profil1d(), and wrout().

5.18.1.62 ireflect

```
integer, dimension(:), allocatable vmec_main::ireflect
```

two-dimensional array for computing 2pi-v angle

Definition at line 141 of file vmec_main.f90.

Referenced by alias(), allocate_ns(), free_mem_ns(), fsym_invfft(), profil3d(), symforce(), symoutput(), and symrzl().

5.18.1.63 irst

```
integer vmec_main::irst
```

"counter" monitoring sign of jacobian; resets R, Z, and Lambda when jacobian changes sign and decreases time step

Definition at line 152 of file vmec_main.f90.

Referenced by eqsolve(), evolve(), fileout(), funct3d(), initialize_radial(), jacobian(), reset_params(), and restart_↵ iter().

5.18.1.64 irzloff

```
integer vmec_main::irzloff
```

offset in xc array between R,Z,L components

Definition at line 149 of file vmec_main.f90.

Referenced by bcovar(), eqsolve(), funct3d(), initialize_radial(), printout(), profil3d(), and wrout().

5.18.1.65 iter1

```
integer vmec_main::iter1
```

number of iterations at which the currently active evolution was branched off from

Definition at line 155 of file vmec_main.f90.

Referenced by bcovar(), eqsolve(), evolve(), funct3d(), initialize_radial(), reset_params(), residue(), and restart_↵ iter().

5.18.1.66 iter2

```
integer vmec_main::iter2
```

total number of iterations

Definition at line 156 of file vmec_main.f90.

Referenced by bcovar(), constrain_m1(), eqsolve(), evolve(), funct3d(), initialize_radial(), printout(), reset_params(), residue(), restart_iter(), and wrout().

5.18.1.67 itfsq

```
integer vmec_main::itfsq
```

Definition at line 143 of file vmec_main.f90.

Referenced by wrout().

5.18.1.68 ivac

```
integer vmec_main::ivac
```

counts number of free-boundary iterations

Definition at line 157 of file vmec_main.f90.

Referenced by eqfor(), eqsolve(), forces(), freeb_data(), funct3d(), printout(), reset_params(), scalfor(), tomnspa(), tomnsp(), and vmec().

5.18.1.69 jcuru

```
real(rprec), dimension(:), allocatable vmec_main::jcuru
```

Definition at line 29 of file vmec_main.f90.

Referenced by allocate_ns(), calc_fbal(), eqfor(), free_mem_ns(), and wrout().

5.18.1.70 jcurv

```
real(rprec), dimension(:), allocatable vmec_main::jcurv
```

Definition at line 30 of file vmec_main.f90.

Referenced by allocate_ns(), calc_fbal(), eqfor(), free_mem_ns(), and wrout().

5.18.1.71 jdotb

```
real(rprec), dimension(:), allocatable vmec_main::jdotb
```

Definition at line 31 of file vmec_main.f90.

Referenced by allocate_ns(), eqfor(), free_mem_ns(), jxbforce(), and wrout().

5.18.1.72 jpar2

```
real(rprec), dimension(:), allocatable vmec_main::jpar2
```

Definition at line 49 of file vmec_main.f90.

Referenced by allocate_ns(), eqfor(), free_mem_ns(), and jxbforce().

5.18.1.73 jperp2

```
real(rprec), dimension(:), allocatable vmec_main::jperp2
```

Definition at line 50 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, `free_mem_ns()`, and `jxbforce()`.

5.18.1.74 lconm1

```
logical vmec_main::lconm1
```

Definition at line 137 of file vmec_main.f90.

Referenced by `constrain_m1()`, `convert_asym()`, `convert_sym()`, `readin()`, `reset_params()`, and `scale_m1()`.

5.18.1.75 lflip

```
logical vmec_main::lflip
```

from `init_geometry`

Definition at line 139 of file vmec_main.f90.

Referenced by `profil1d()`, and `readin()`.

5.18.1.76 lthreed

```
logical vmec_main::lthreed
```

Definition at line 136 of file vmec_main.f90.

Referenced by `bcovar()`, `convert()`, `eqfor()`, `flip_theta()`, `forces()`, `read_indata()`, `readin()`, `residue()`, `spectrum()`, `symforce()`, `symrzi()`, `tomnspa()`, `tomnsp()`, `totzspa()`, `totzsps()`, and `wrout()`.

5.18.1.77 mass

```
real(rprec), dimension(:), allocatable vmec_main::mass
```

mass profile on half-grid

Definition at line 67 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `free_mem_ns()`, `profil1d()`, and `wrout()`.

5.18.1.78 multi_ns_grid

```
integer vmec_main::multi_ns_grid
```

Definition at line 142 of file vmec_main.f90.

Referenced by readin(), and vmec().

5.18.1.79 ndatai

```
integer vmec_main::ndatai
```

Definition at line 145 of file vmec_main.f90.

5.18.1.80 ndatap

```
integer vmec_main::ndatap
```

Definition at line 144 of file vmec_main.f90.

5.18.1.81 neqs

```
integer vmec_main::neqs
```

total number of equations to evolve (size of xc)

Definition at line 148 of file vmec_main.f90.

Referenced by allocate_ns(), funct3d(), initialize_radial(), restart_iter(), and wrout().

5.18.1.82 niterv

```
integer vmec_main::niterv
```

max iterations for current multi-grid iteration

Definition at line 146 of file vmec_main.f90.

Referenced by eqsolve(), initialize_radial(), and vmec().

5.18.1.83 ohs

```
real(rprec) vmec_main::ohs
```

Definition at line 86 of file vmec_main.f90.

Referenced by calc_fbal(), forces(), funct3d(), initialize_radial(), jacobian(), jxbforce(), and precondition().

5.18.1.84 otau

```
real(rprec), dimension(ndamp) vmec_main::otau
```

Definition at line 113 of file vmec_main.f90.

Referenced by evolve().

5.18.1.85 otav

```
real(rprec) vmec_main::otav
```

time-step algorithm

Definition at line 112 of file vmec_main.f90.

Referenced by evolve().

5.18.1.86 overr

```
real(rprec), dimension(:), allocatable vmec_main::overr
```

Definition at line 44 of file vmec_main.f90.

Referenced by allocate_ns(), eqfor(), free_mem_ns(), and wrout().

5.18.1.87 phi

```
real(rprec), dimension(:), allocatable vmec_main::phi
```

Definition at line 27 of file vmec_main.f90.

Referenced by allocate_ns(), fileout(), free_mem_ns(), jxbforce(), and wrout().

5.18.1.88 phipf

```
real(rprec), dimension(:), allocatable vmec_main::phipf
```

Definition at line 25 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `calc_fbal()`, `eqfor()`, `free_mem_ns()`, `profil1d()`, and `wrout()`.

5.18.1.89 phips

```
real(rprec), dimension(:), allocatable vmec_main::phips
```

toroidal flux (same as `phip`), one-dimensional array

Definition at line 64 of file vmec_main.f90.

Referenced by `add_fluxes()`, `allocate_ns()`, `free_mem_ns()`, `jxbforce()`, `profil1d()`, `profil3d()`, and `wrout()`.

5.18.1.90 pres

```
real(rprec), dimension(:), allocatable vmec_main::pres
```

Definition at line 47 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `calc_fbal()`, `eqfor()`, `free_mem_ns()`, `funct3d()`, `jxbforce()`, `profil1d()`, and `wrout()`.

5.18.1.91 presf

```
real(rprec), dimension(:), allocatable vmec_main::presf
```

pressure profile on full-grid, $\text{mass}/\text{phip} \times \gamma$

Definition at line 62 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, `free_mem_ns()`, and `wrout()`.

5.18.1.92 presgrad

```
real(rprec), dimension(:), allocatable vmec_main::presgrad
```

Definition at line 56 of file vmec_main.f90.

Referenced by `allocate_ns()`, `calc_fbal()`, `eqfor()`, and `free_mem_ns()`.

5.18.1.93 psi

```
real(rprec), dimension(:), allocatable vmec_main::psi
```

Definition at line 38 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, and `free_mem_ns()`.

5.18.1.94 r00

```
real(rprec) vmec_main::r00
```

Definition at line 88 of file vmec_main.f90.

Referenced by `eqfor()`, `funct3d()`, `printout()`, and `profil1d()`.

5.18.1.95 r0scale

```
real(rprec) vmec_main::r0scale
```

Definition at line 89 of file vmec_main.f90.

Referenced by `bcovar()`, `fixaray()`, `jxbforce()`, `lamcal()`, `precondn()`, `residue()`, and `wrout()`.

5.18.1.96 rbsq

```
real(rprec), dimension(:), allocatable vmec_main::rbsq
```

Definition at line 120 of file vmec_main.f90.

Referenced by `allocate_nunv()`, `forces()`, `free_mem_nunv()`, and `funct3d()`.

5.18.1.97 rbtor

```
real(rprec) vmec_main::rbtor
```

Definition at line 122 of file vmec_main.f90.

Referenced by `bcovar()`, `eqfor()`, `funct3d()`, and `wrout()`.

5.18.1.98 rbtor0

```
real(rprec) vmec_main::rbtor0
```

Definition at line 123 of file vmec_main.f90.

Referenced by bcovar(), eqfor(), and wrout().

5.18.1.99 res0

```
real(rprec) vmec_main::res0
```

Definition at line 126 of file vmec_main.f90.

Referenced by eqsolve(), initialize_radial(), and reset_params().

5.18.1.100 rinner

```
real(rprec) vmec_main::rinner
```

Definition at line 107 of file vmec_main.f90.

Referenced by eqfor(), and funct3d().

5.18.1.101 rmn_bdy

```
real(rprec), dimension(:,:,:), allocatable, target vmec_main::rmn_bdy
```

Definition at line 115 of file vmec_main.f90.

Referenced by allocate_nunv(), free_mem_nunv(), profil1d(), profil3d(), and readin().

5.18.1.102 router

```
real(rprec) vmec_main::router
```

Definition at line 106 of file vmec_main.f90.

Referenced by eqfor(), and funct3d().

5.18.1.103 sifa

```
real(rprec), dimension(ndatafmax) vmec_main::sifa
```

Definition at line 132 of file vmec_main.f90.

5.18.1.104 sifa2

```
real(rprec), dimension(ndatafmax) vmec_main::sifa2
```

Definition at line 133 of file vmec_main.f90.

5.18.1.105 sm

```
real(rprec), dimension(:), allocatable vmec_main::sm
```

Definition at line 45 of file vmec_main.f90.

Referenced by `allocate_ns()`, `free_mem_ns()`, `precondn()`, `profil1d()`, and `wrout()`.

5.18.1.106 sp

```
real(rprec), dimension(:), allocatable vmec_main::sp
```

Definition at line 46 of file vmec_main.f90.

Referenced by `allocate_ns()`, `free_mem_ns()`, `precondn()`, `profil1d()`, and `wrout()`.

5.18.1.107 specw

```
real(rprec), dimension(:), allocatable vmec_main::specw
```

Definition at line 36 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, `free_mem_ns()`, `printout()`, `spectrum()`, and `wrout()`.

5.18.1.108 spfa

```
real(rprec), dimension(ndatafmax) vmec_main::spfa
```

Definition at line 129 of file vmec_main.f90.

5.18.1.109 spfa2

```
real(rprec), dimension(ndatafmax) vmec_main::spfa2
```

Definition at line 130 of file vmec_main.f90.

5.18.1.110 tcon

```
real(rprec), dimension(:), allocatable vmec_main::tcon
```

Definition at line 37 of file vmec_main.f90.

Referenced by alias(), allocate_ns(), bcovar(), and free_mem_ns().

5.18.1.111 vacuum_calls

```
integer vmec_main::vacuum_calls = 0
```

Definition at line 159 of file vmec_main.f90.

Referenced by funct3d(), and reset_params().

5.18.1.112 voli

```
real(rprec) vmec_main::voli
```

Definition at line 87 of file vmec_main.f90.

Referenced by bcovar().

5.18.1.113 vp

```
real(rprec), dimension(:), allocatable vmec_main::vp
```

Definition at line 48 of file vmec_main.f90.

Referenced by `allocate_ns()`, `bcovar()`, `calc_fbal()`, `eqfor()`, `free_mem_ns()`, `jxbforce()`, `precondn()`, `printout()`, and `wrout()`.

5.18.1.114 vpphi

```
real(rprec), dimension(:), allocatable vmec_main::vpphi
```

Definition at line 55 of file vmec_main.f90.

Referenced by `allocate_ns()`, `calc_fbal()`, `eqfor()`, and `free_mem_ns()`.

5.18.1.115 wb

```
real(rprec) vmec_main::wb
```

Definition at line 103 of file vmec_main.f90.

Referenced by `bcovar()`, `eqsolve()`, `printout()`, and `wrout()`.

5.18.1.116 wp

```
real(rprec) vmec_main::wp
```

Definition at line 104 of file vmec_main.f90.

Referenced by `bcovar()`, `eqsolve()`, `printout()`, and `wrout()`.

5.18.1.117 xcl0

```
real(rprec), dimension(:), allocatable vmec_main::xcl0
```

Definition at line 78 of file vmec_main.f90.

5.18.1.118 xmpq

```
real(rprec), dimension(0:mpolld,3) vmec_main::xmpq
```

Definition at line 80 of file vmec_main.f90.

Referenced by fixaray(), spectrum(), tomnspa(), tomnsps(), totzspa(), and totzsps().

5.18.1.119 yellip

```
real(rprec), dimension(:), allocatable vmec_main::yellip
```

Definition at line 39 of file vmec_main.f90.

Referenced by allocate_ns(), eqfor(), and free_mem_ns().

5.18.1.120 ygeo

```
real(rprec), dimension(:), allocatable vmec_main::ygeo
```

Definition at line 43 of file vmec_main.f90.

Referenced by allocate_ns(), eqfor(), and free_mem_ns().

5.18.1.121 yinden

```
real(rprec), dimension(:), allocatable vmec_main::yinden
```

Definition at line 40 of file vmec_main.f90.

Referenced by allocate_ns(), eqfor(), and free_mem_ns().

5.18.1.122 yshift

```
real(rprec), dimension(:), allocatable vmec_main::yshift
```

Definition at line 42 of file vmec_main.f90.

Referenced by allocate_ns(), eqfor(), and free_mem_ns().

5.18.1.123 ytrian

```
real(rprec), dimension(:), allocatable vmec_main::ytrian
```

Definition at line 41 of file vmec_main.f90.

Referenced by `allocate_ns()`, `eqfor()`, and `free_mem_ns()`.

5.18.1.124 z00

```
real(rprec) vmec_main::z00
```

Definition at line 90 of file vmec_main.f90.

Referenced by `funct3d()`, `printout()`, and `reset_params()`.

5.18.1.125 zmn_bdy

```
real(rprec), dimension(:,:), allocatable, target vmec_main::zmn_bdy
```

Definition at line 116 of file vmec_main.f90.

Referenced by `allocate_nunv()`, `free_mem_nunv()`, `profil3d()`, and `readin()`.

5.19 vmec_params Module Reference

Variables

- integer, parameter `meven` = 0
parity selection label for even poloidal modes of R and Z
- integer, parameter `modd` = 1
parity selection label for odd poloidal modes of R and Z
- integer, parameter `ndamp` = 10
number of iterations over which damping is averaged
- integer, parameter `ns4` = 25
- integer, dimension(0:mpold), parameter `jmin1` = (/ 1,1,(2,ink=2,mpold) /)
starting js(m) values where R,Z are non-zero
- integer, dimension(0:mpold), parameter `jmin2` = (/ 1,2,(2,ink=2,mpold) /)
starting js(m) values for which R,Z are evolved
- integer, dimension(0:mpold), parameter `jlam` = (/ 2,2,(2,ink=2,mpold) /)
starting js(m) values for which Lambda is evolved
- integer, parameter `norm_term_flag` = 0
- integer, parameter `bad_jacobian_flag` = 1
- integer, parameter `jac75_flag` = 4
- integer, parameter `input_error_flag` = 5
- integer, parameter `phiedge_error_flag` = 7

- integer, parameter `ns_error_flag` = 8
- integer, parameter `misc_error_flag` = 9
- integer, parameter `successful_term_flag` = 11
- integer, parameter `restart_flag` = 1
- integer, parameter `readin_flag` = 2
- integer, parameter `timestep_flag` = 4
- integer, parameter `output_flag` = 8
- integer, parameter `cleanup_flag` = 16
- integer, parameter `reset_jacdt_flag` = 32
- real(rprec), parameter `pdamp` = 0.05_dp
- character(len= *), parameter `version_` = '8.52'
- integer `ntmax`
number of contributing Fourier basis function (can be 1, 2 or 4); assigned in `read_indata()`
- integer `rcc`
- integer `rss`
- integer `rsc`
- integer `rzs`
- integer `zsc`
- integer `zcs`
- integer `zcc`
- integer `zss`
- integer `mnyq`
- integer `nnyq`
- integer, dimension(:), allocatable `uminus`
- real(rprec), dimension(:), allocatable `mscale`
array for norming theta-trig functions (internal use only) so that the discrete $SUM[\cos(\mu)\cos(m'u)] = .5 \delta(m,m')$*
- real(rprec), dimension(:), allocatable `nscale`
array for norming zeta -trig functions (internal use only)
- real(rprec) `signgs`
sign of Jacobian : must be =1 (right-handed) or =-1 (left-handed)
- real(rprec) `lamscale` =1
- integer, parameter `m0` =0
from totzsp
- integer, parameter `m1` =1
from totzsp
- integer, parameter `n0` =0
from totzsp

5.19.1 Variable Documentation

5.19.1.1 bad_jacobian_flag

integer, parameter `vmec_params::bad_jacobian_flag` = 1

Definition at line 20 of file `vmec_params.f90`.

Referenced by `eqsolve()`, `evolve()`, `funct3d()`, and `vmec()`.

5.19.1.2 cleanup_flag

```
integer, parameter vmec_params::cleanup_flag = 16
```

Definition at line 32 of file vmec_params.f90.

5.19.1.3 input_error_flag

```
integer, parameter vmec_params::input_error_flag = 5
```

Definition at line 22 of file vmec_params.f90.

Referenced by read_indata().

5.19.1.4 jac75_flag

```
integer, parameter vmec_params::jac75_flag = 4
```

Definition at line 21 of file vmec_params.f90.

Referenced by eqsolve().

5.19.1.5 jlam

```
integer, dimension(0:mpold), parameter vmec_params::jlam = (/ 2,2,(2,ink=2,mpold) /)
```

starting js(m) values for which Lambda is evolved

Definition at line 17 of file vmec_params.f90.

Referenced by lamcal(), tomnsps(), tomnsps(), totzsps(), and totzsps().

5.19.1.6 jmin1

```
integer, dimension(0:mpold), parameter vmec_params::jmin1 = (/ 1,1,(2,ink=2,mpold) /)
```

starting js(m) values where R,Z are non-zero

Definition at line 15 of file vmec_params.f90.

Referenced by totzsps(), and totzsps().

5.19.1.7 jmin2

```
integer, dimension(0:mpold), parameter vmec_params::jmin2 = (/ 1,2, (2,ink=2,mpold) /)
```

starting js(m) values for which R,Z are evolved

Definition at line 16 of file vmec_params.f90.

Referenced by fixaray(), scalfor(), tomnspace(), and tomnsps().

5.19.1.8 lamscale

```
real(rprec) vmec_params::lamscale =1
```

Definition at line 58 of file vmec_params.f90.

Referenced by bcovar(), lamcal(), profil1d(), and wrout().

5.19.1.9 m0

```
integer, parameter vmec_params::m0 =0
```

from totzsp

Definition at line 60 of file vmec_params.f90.

Referenced by totzspace(), and totzsps().

5.19.1.10 m1

```
integer, parameter vmec_params::m1 =1
```

from totzsp

Definition at line 61 of file vmec_params.f90.

Referenced by convert_sym(), totzspace(), and totzsps().

5.19.1.11 meven

```
integer, parameter vmec_params::meven = 0
```

parity selection label for even poloidal modes of R and Z

Definition at line 9 of file vmec_params.f90.

Referenced by jacobian(), and residue().

5.19.1.12 misc_error_flag

```
integer, parameter vmec_params::misc_error_flag = 9
```

Definition at line 25 of file vmec_params.f90.

5.19.1.13 mnyq

```
integer vmec_params::mnyq
```

Definition at line 49 of file vmec_params.f90.

Referenced by fixaray(), jxbforce(), and wrout().

5.19.1.14 modd

```
integer, parameter vmec_params::modd = 1
```

parity selection label for odd poloidal modes of R and Z

Definition at line 10 of file vmec_params.f90.

Referenced by jacobian(), and residue().

5.19.1.15 mscale

```
real(rprec), dimension(:), allocatable vmec_params::mscale
```

array for norming theta-trig functions (internal use only) so that the discrete $\text{SUM}[\cos(\mu) \cdot \cos(m'u)] = .5 \delta(m, m')$

Definition at line 53 of file vmec_params.f90.

Referenced by convert(), eqfor(), fixaray(), profil3d(), spectrum(), vmec(), and wrout().

5.19.1.16 n0

```
integer, parameter vmec_params::n0 =0
```

from totzsp

Definition at line 62 of file vmec_params.f90.

Referenced by totzspa(), and totzsps().

5.19.1.17 ndamp

```
integer, parameter vmec_params::ndamp = 10
```

number of iterations over which damping is averaged

Definition at line 11 of file vmec_params.f90.

5.19.1.18 nnyq

```
integer vmec_params::nnyq
```

Definition at line 50 of file vmec_params.f90.

Referenced by fixaray(), jxbforce(), and wrout().

5.19.1.19 norm_term_flag

```
integer, parameter vmec_params::norm_term_flag = 0
```

Definition at line 19 of file vmec_params.f90.

Referenced by eqsolve(), evolve(), fileout(), readin(), vacuum(), and vmec().

5.19.1.20 ns4

```
integer, parameter vmec_params::ns4 = 25
```

Definition at line 12 of file vmec_params.f90.

Referenced by bcovar(), and eqsolve().

5.19.1.21 ns_error_flag

```
integer, parameter vmec_params::ns_error_flag = 8
```

Definition at line 24 of file vmec_params.f90.

5.19.1.22 nscale

```
real(rprec), dimension(:), allocatable vmec_params::nscale
```

array for norming zeta -trig functions (internal use only)

Definition at line 55 of file vmec_params.f90.

Referenced by `convert()`, `eqfor()`, `fixarray()`, `guess_axis()`, `profil3d()`, `spectrum()`, `vmec()`, and `wrout()`.

5.19.1.23 ntmax

```
integer vmec_params::ntmax
```

number of contributing Fourier basis function (can be 1, 2 or 4); assigned in `read_indata()`

Definition at line 38 of file vmec_params.f90.

Referenced by `allocate_ns()`, `allocate_nunv()`, `convert()`, `eqfor()`, `flip_theta()`, `getfsq()`, `initialize_radial()`, `interp()`, `lamcal()`, `profil3d()`, `read_indata()`, `residue()`, `scalfor()`, `spectrum()`, `tomnspa()`, `tomnsps()`, `totzspa()`, `totzsps()`, and `wrout()`.

5.19.1.24 output_flag

```
integer, parameter vmec_params::output_flag = 8
```

Definition at line 31 of file vmec_params.f90.

5.19.1.25 pdamp

```
real(rprec), parameter vmec_params::pdamp = 0.05_dp
```

Definition at line 35 of file vmec_params.f90.

Referenced by `bcovar()`, and `profil1d()`.

5.19.1.26 phiedge_error_flag

```
integer, parameter vmec_params::phiedge_error_flag = 7
```

Definition at line 23 of file vmec_params.f90.

Referenced by vacuum().

5.19.1.27 rcc

```
integer vmec_params::rcc
```

Definition at line 40 of file vmec_params.f90.

Referenced by convert(), eqfor(), flip_theta(), profil1d(), profil3d(), read_indata(), readin(), tomnsps(), and totzsps().

5.19.1.28 rcs

```
integer vmec_params::rcs
```

Definition at line 43 of file vmec_params.f90.

Referenced by convert(), eqfor(), flip_theta(), profil3d(), read_indata(), readin(), tomnsps(), and totzsps().

5.19.1.29 readin_flag

```
integer, parameter vmec_params::readin_flag = 2
```

Definition at line 29 of file vmec_params.f90.

5.19.1.30 reset_jacdt_flag

```
integer, parameter vmec_params::reset_jacdt_flag = 32
```

Definition at line 33 of file vmec_params.f90.

5.19.1.31 restart_flag

```
integer, parameter vmec_params::restart_flag = 1
```

Definition at line 28 of file vmec_params.f90.

5.19.1.32 rsc

```
integer vmec_params::rsc
```

Definition at line 42 of file vmec_params.f90.

Referenced by `convert()`, `flip_theta()`, `read_indata()`, `readin()`, `residue()`, `spectrum()`, `tomnspa()`, `totzspa()`, and `wrout()`.

5.19.1.33 rss

```
integer vmec_params::rss
```

Definition at line 41 of file vmec_params.f90.

Referenced by `convert()`, `flip_theta()`, `read_indata()`, `readin()`, `residue()`, `spectrum()`, `tomnsps()`, `totzsps()`, and `wrout()`.

5.19.1.34 signgs

```
real(rprec) vmec_params::signgs
```

sign of Jacobian : must be =1 (right-handed) or =-1 (left-handed)

Definition at line 56 of file vmec_params.f90.

Referenced by `bcovar()`, `calc_fbal()`, `eqfor()`, `fileout()`, `fixaray()`, `funct3d()`, `guess_axis()`, `jxbforce()`, `precondn()`, `profil1d()`, `readin()`, and `wrout()`.

5.19.1.35 successful_term_flag

```
integer, parameter vmec_params::successful_term_flag = 11
```

Definition at line 26 of file vmec_params.f90.

Referenced by `eqsolve()`, `evolve()`, `fileout()`, `jxbforce()`, and `vmec()`.

5.19.1.36 timestep_flag

```
integer, parameter vmec_params::timestep_flag = 4
```

Definition at line 30 of file vmec_params.f90.

5.19.1.37 uminus

```
integer, dimension(:), allocatable vmec_params::uminus
```

Definition at line 52 of file vmec_params.f90.

Referenced by profil3d(), and vmec().

5.19.1.38 version_

```
character(len=*), parameter vmec_params::version_ = '8.52'
```

Definition at line 36 of file vmec_params.f90.

Referenced by heading(), and wrout().

5.19.1.39 zcc

```
integer vmec_params::zcc
```

Definition at line 46 of file vmec_params.f90.

Referenced by convert(), eqfor(), flip_theta(), profil3d(), read_indata(), readin(), residue(), spectrum(), tomnspa(), totzspa(), and wrout().

5.19.1.40 zcs

```
integer vmec_params::zcs
```

Definition at line 45 of file vmec_params.f90.

Referenced by convert(), eqfor(), flip_theta(), profil3d(), read_indata(), readin(), residue(), spectrum(), tomnsp(), totzsps(), and wrout().

5.19.1.41 zsc

```
integer vmec_params::zsc
```

Definition at line 44 of file vmec_params.f90.

Referenced by `convert()`, `eqfor()`, `flip_theta()`, `read_indata()`, `readin()`, `tomnsps()`, and `totzsps()`.

5.19.1.42 zss

```
integer vmec_params::zss
```

Definition at line 47 of file vmec_params.f90.

Referenced by `convert()`, `flip_theta()`, `read_indata()`, `readin()`, `tomnsps()`, and `totzsps()`.

5.20 vmec_persistent Module Reference

Variables

- integer, dimension(:), allocatable [ixm](#)
- integer, dimension(:), allocatable [jmin3](#)
- real(rprec), dimension(:,:), allocatable [cosmu](#)
- real(rprec), dimension(:,:), allocatable [sinmu](#)
- real(rprec), dimension(:,:), allocatable [cosmum](#)
- real(rprec), dimension(:,:), allocatable [sinmum](#)
- real(rprec), dimension(:,:), allocatable [cosmumi](#)
- real(rprec), dimension(:,:), allocatable [sinmumi](#)
- real(rprec), dimension(:,:), allocatable [cosnv](#)
- real(rprec), dimension(:,:), allocatable [sinnv](#)
- real(rprec), dimension(:,:), allocatable [cosnvn](#)
- real(rprec), dimension(:,:), allocatable [sinnav](#)
- real(rprec), dimension(:,:), allocatable [cosmui](#)
- real(rprec), dimension(:,:), allocatable [sinmui](#)
- real(rprec), dimension(:,:), allocatable [cosmui3](#)
- real(rprec), dimension(:,:), allocatable [cosmumi3](#)
- real(rprec), dimension(:), allocatable, target [xm](#)
- real(rprec), dimension(:), allocatable, target [xn](#)
- real(rprec), dimension(:), allocatable, target [xm_nyq](#)
- real(rprec), dimension(:), allocatable, target [xn_nyq](#)
- real(rprec), dimension(:), allocatable [cos01](#)
- real(rprec), dimension(:), allocatable [sin01](#)

5.20.1 Variable Documentation

5.20.1.1 cos01

```
real(rprec), dimension(:), allocatable vmec_persistent::cos01
```

Definition at line 32 of file vmec_persistent.f90.

5.20.1.2 cosmu

```
real(rprec), dimension(:, :), allocatable vmec_persistent::cosmu
```

Definition at line 11 of file vmec_persistent.f90.

Referenced by getbsubs().

5.20.1.3 cosmui

```
real(rprec), dimension(:, :), allocatable vmec_persistent::cosmui
```

Definition at line 22 of file vmec_persistent.f90.

Referenced by wrout().

5.20.1.4 cosmui3

```
real(rprec), dimension(:, :), allocatable vmec_persistent::cosmui3
```

Definition at line 24 of file vmec_persistent.f90.

5.20.1.5 cosmum

```
real(rprec), dimension(:, :), allocatable vmec_persistent::cosmum
```

Definition at line 13 of file vmec_persistent.f90.

5.20.1.6 cosmumi

```
real(rprec), dimension(:, :), allocatable vmec_persistent::cosmumi
```

Definition at line 15 of file vmec_persistent.f90.

5.20.1.7 **cosmumi3**

```
real(rprec), dimension(:,:), allocatable vmec_persistent::cosmumi3
```

Definition at line 25 of file vmec_persistent.f90.

5.20.1.8 **cosnv**

```
real(rprec), dimension(:,:), allocatable vmec_persistent::cosnv
```

Definition at line 17 of file vmec_persistent.f90.

Referenced by getbsubs(), and wrout().

5.20.1.9 **cosnvn**

```
real(rprec), dimension(:,:), allocatable vmec_persistent::cosnvn
```

Definition at line 19 of file vmec_persistent.f90.

5.20.1.10 **ixm**

```
integer, dimension(:), allocatable vmec_persistent::ixm
```

Definition at line 8 of file vmec_persistent.f90.

Referenced by interp().

5.20.1.11 **jmin3**

```
integer, dimension(:), allocatable vmec_persistent::jmin3
```

Definition at line 9 of file vmec_persistent.f90.

5.20.1.12 **sin01**

```
real(rprec), dimension(:), allocatable vmec_persistent::sin01
```

Definition at line 33 of file vmec_persistent.f90.

5.20.1.13 sinmu

```
real(rprec), dimension(:,:), allocatable vmec_persistent::sinmu
```

Definition at line 12 of file vmec_persistent.f90.

Referenced by getbsubs().

5.20.1.14 sinmui

```
real(rprec), dimension(:,:), allocatable vmec_persistent::sinmui
```

Definition at line 23 of file vmec_persistent.f90.

Referenced by wrout().

5.20.1.15 sinmum

```
real(rprec), dimension(:,:), allocatable vmec_persistent::sinmum
```

Definition at line 14 of file vmec_persistent.f90.

5.20.1.16 sinmumi

```
real(rprec), dimension(:,:), allocatable vmec_persistent::sinmumi
```

Definition at line 16 of file vmec_persistent.f90.

5.20.1.17 sinnv

```
real(rprec), dimension(:,:), allocatable vmec_persistent::sinnv
```

Definition at line 18 of file vmec_persistent.f90.

Referenced by getbsubs(), and wrout().

5.20.1.18 sinnvn

```
real(rprec), dimension(:,:), allocatable vmec_persistent::sinnvn
```

Definition at line 20 of file vmec_persistent.f90.

5.20.1.19 xm

```
real(rprec), dimension(:), allocatable, target vmec_persistent::xm
```

Definition at line 27 of file vmec_persistent.f90.

Referenced by wrout().

5.20.1.20 xm_nyq

```
real(rprec), dimension(:), allocatable, target vmec_persistent::xm_nyq
```

Definition at line 29 of file vmec_persistent.f90.

Referenced by wrout().

5.20.1.21 xn

```
real(rprec), dimension(:), allocatable, target vmec_persistent::xn
```

Definition at line 28 of file vmec_persistent.f90.

Referenced by wrout().

5.20.1.22 xn_nyq

```
real(rprec), dimension(:), allocatable, target vmec_persistent::xn_nyq
```

Definition at line 30 of file vmec_persistent.f90.

Referenced by wrout().

5.21 vmercier Module Reference

Variables

- `real(rprec), dimension(nsd)` [dshear](#)
- `real(rprec), dimension(nsd)` [dwell](#)
- `real(rprec), dimension(nsd)` [dcurr](#)
- `real(rprec), dimension(nsd)` [dmerc](#)
- `real(rprec), dimension(nsd)` [dgeod](#)

5.21.1 Variable Documentation

5.21.1.1 dcurr

```
real(rprec), dimension(nsd) vmercier::dcurr
```

Definition at line 10 of file vmercier.f90.

Referenced by `mercier()`, and `wrout()`.

5.21.1.2 dgeod

```
real(rprec), dimension(nsd) vmercier::dgeod
```

Definition at line 12 of file vmercier.f90.

Referenced by `mercier()`, and `wrout()`.

5.21.1.3 dmerc

```
real(rprec), dimension(nsd) vmercier::dmerc
```

Definition at line 11 of file vmercier.f90.

Referenced by `mercier()`, and `wrout()`.

5.21.1.4 dshear

```
real(rprec), dimension(nsd) vmercier::dshear
```

Definition at line 8 of file vmercier.f90.

Referenced by mercier(), and wrout().

5.21.1.5 dwell

```
real(rprec), dimension(nsd) vmercier::dwell
```

Definition at line 9 of file vmercier.f90.

Referenced by mercier(), and wrout().

5.22 vparams Module Reference

Variables

- integer, parameter `nsd` = 10001
maximum number of radial nodes
- integer, parameter `mpold` = 101
maximum number of poloidal harmonics (in r,z,lam fourier series)
- integer, parameter `ntord` = 101
maximum number of toroidal harmonics
- integer, parameter `ndatafmax` = 101
- integer, parameter `nstore_seq` = 100
- integer, parameter `mpol1d` = `mpold` - 1
- integer, parameter `ntor1d` = `ntord` + 1
- integer, parameter `nthreed0` = 9
- integer, parameter `indata0` = `nthreed0` + 2
- integer, parameter `nwout0` = `nthreed0` + 3
- integer, parameter `jxbout0` = `nthreed0` + 4
- integer, parameter `nfort18` = 18
- integer, parameter `nmercier0` = 52
- integer `nthreed`
- real(rprec), parameter `c1pm2` = 1.e-2_dp
- real(rprec), parameter `cp15` = 0.15_dp
- real(rprec), parameter `cp25` = 0.25_dp
- real(rprec), parameter `cp5` = 0.50_dp
- real(rprec), parameter `c1pm8` = 1.0e-8_dp
- real(rprec), parameter `cbig` = 0.9e30_dp
- real(rprec), parameter `c2p0` = 2
- real(rprec), parameter `c3p0` = 3
- real(rprec), parameter `cp05` = 0.05_dp
- real(rprec), parameter `c1pm13` = 1.0e-13_dp
- real(rprec), parameter `osqrt2` = 0.707106781186547462_dp

5.22.1 Variable Documentation

5.22.1.1 c1pm13

```
real(rprec), parameter vparams::c1pm13 = 1.0e-13_dp
```

Definition at line 41 of file vparams.f90.

5.22.1.2 c1pm2

```
real(rprec), parameter vparams::c1pm2 = 1.e-2_dp
```

Definition at line 32 of file vparams.f90.

5.22.1.3 c1pm8

```
real(rprec), parameter vparams::c1pm8 = 1.0e-8_dp
```

Definition at line 36 of file vparams.f90.

5.22.1.4 c2p0

```
real(rprec), parameter vparams::c2p0 = 2
```

Definition at line 38 of file vparams.f90.

Referenced by wrout().

5.22.1.5 c3p0

```
real(rprec), parameter vparams::c3p0 = 3
```

Definition at line 39 of file vparams.f90.

5.22.1.6 cbig

```
real(rprec), parameter vparams::cbig = 0.9e30_dp
```

Definition at line 37 of file vparams.f90.

5.22.1.7 cp05

```
real(rprec), parameter vparams::cp05 = 0.05_dp
```

Definition at line 40 of file vparams.f90.

5.22.1.8 cp15

```
real(rprec), parameter vparams::cp15 = 0.15_dp
```

Definition at line 33 of file vparams.f90.

5.22.1.9 cp25

```
real(rprec), parameter vparams::cp25 = 0.25_dp
```

Definition at line 34 of file vparams.f90.

5.22.1.10 cp5

```
real(rprec), parameter vparams::cp5 = 0.50_dp
```

Definition at line 35 of file vparams.f90.

Referenced by wrout().

5.22.1.11 indata0

```
integer, parameter vparams::indata0 = nthreed0 + 2
```

Definition at line 24 of file vparams.f90.

5.22.1.12 jxbout0

```
integer, parameter vparams::jxbout0 = nthreed0 + 4
```

Definition at line 26 of file vparams.f90.

5.22.1.13 mpold

```
integer, parameter vparams::mpold = mpold - 1
```

Definition at line 19 of file vparams.f90.

5.22.1.14 mpold

```
integer, parameter vparams::mpold = 101
```

maximum number of poloidal harmonics (in r,z,lam fourier series)

Definition at line 13 of file vparams.f90.

5.22.1.15 ndatafmax

```
integer, parameter vparams::ndatafmax = 101
```

Definition at line 15 of file vparams.f90.

5.22.1.16 nfort18

```
integer, parameter vparams::nfort18 = 18
```

Definition at line 27 of file vparams.f90.

5.22.1.17 nmercier0

```
integer, parameter vparams::nmercier0 = 52
```

Definition at line 28 of file vparams.f90.

Referenced by mercier().

5.22.1.18 nsd

```
integer, parameter vparams::nsd = 10001
```

maximum number of radial nodes

Definition at line 12 of file vparams.f90.

5.22.1.19 nstore_seq

```
integer, parameter vparams::nstore_seq = 100
```

Definition at line 16 of file vparams.f90.

5.22.1.20 nthreed

```
integer vparams::nthreed
```

Definition at line 29 of file vparams.f90.

Referenced by heading(), open_output_files(), and vmec().

5.22.1.21 nthreed0

```
integer, parameter vparams::nthreed0 = 9
```

Definition at line 23 of file vparams.f90.

Referenced by open_output_files().

5.22.1.22 ntor1d

```
integer, parameter vparams::ntor1d = ntord + 1
```

Definition at line 20 of file vparams.f90.

5.22.1.23 ntord

```
integer, parameter vparams::ntord = 101
```

maximum number of toroidal harmonics

Definition at line 14 of file vparams.f90.

5.22.1.24 nwout0

```
integer, parameter vparams::nwout0 = nthreed0 + 3
```

Definition at line 25 of file vparams.f90.

5.22.1.25 osqrt2

```
real(rprec), parameter vparams::osqrt2 = 0.707106781186547462_dp
```

Definition at line 42 of file vparams.f90.

5.23 vsvd0 Module Reference**Variables**

- integer, parameter `nigroup` = 100
number of external current groups

5.23.1 Variable Documentation**5.23.1.1 nigroup**

```
integer, parameter vsvd0::nigroup = 100
```

number of external current groups

Definition at line 8 of file vsvd0.f90.

Referenced by `mgrid_mod::read_mgrid()`.

5.24 xstuff Module Reference

Variables

- `real(rprec), dimension(:), allocatable gc`
stacked array of R, Z, Lambda Spectral force coefficients (see above for stack order)
- `real(rprec), dimension(:), allocatable, target xc`
stacked array of scaled R, Z, Lambda Fourier coefficients (see above for stack order)
- `real(rprec), dimension(:), allocatable xcdot`
"velocity": change of Fourier coefficients per time step
- `real(rprec), dimension(:), allocatable xsave`
- `real(rprec), dimension(:), allocatable xstore`
backup copy of last-known-good xc
- `real(rprec), dimension(:), allocatable scalxc`

5.24.1 Variable Documentation

5.24.1.1 gc

```
real(rprec), dimension(:), allocatable xstuff::gc
```

stacked array of R, Z, Lambda Spectral force coefficients (see above for stack order)

Definition at line 39 of file `xstuff.f90`.

Referenced by `allocate_ns()`, `evolve()`, `fileout()`, `free_mem_ns()`, `funct3d()`, `initialize_radial()`, and `printout()`.

5.24.1.2 scalxc

```
real(rprec), dimension(:), allocatable xstuff::scalxc
```

Definition at line 52 of file `xstuff.f90`.

Referenced by `allocate_ns()`, `free_mem_ns()`, `funct3d()`, `initialize_radial()`, and `profil3d()`.

5.24.1.3 xc

```
real(rprec), dimension(:), allocatable, target xstuff::xc
```

stacked array of scaled R, Z, Lambda Fourier coefficients (see above for stack order)

Definition at line 42 of file `xstuff.f90`.

Referenced by `allocate_ns()`, `bcovar()`, `eqsolve()`, `evolve()`, `fileout()`, `free_mem_ns()`, `funct3d()`, `initialize_radial()`, `restart_iter()`, and `wrout()`.

5.24.1.4 xcdot

```
real(rprec), dimension(:), allocatable xstuff::xcdot
```

"velocity": change of Fourier coefficients per time step

Definition at line 45 of file xstuff.f90.

Referenced by `allocate_ns()`, `evolve()`, `free_mem_ns()`, `initialize_radial()`, and `restart_iter()`.

5.24.1.5 xsave

```
real(rprec), dimension(:), allocatable xstuff::xsave
```

Definition at line 47 of file xstuff.f90.

Referenced by `allocate_ns()`, `fileout()`, and `free_mem_ns()`.

5.24.1.6 xstore

```
real(rprec), dimension(:), allocatable xstuff::xstore
```

backup copy of last-known-good xc

Definition at line 50 of file xstuff.f90.

Referenced by `allocate_ns()`, `free_mem_ns()`, `initialize_radial()`, `printout()`, and `restart_iter()`.

Chapter 6

Data Type Documentation

6.1 read_wout_mod::read_wout_file Interface Reference

Public Member Functions

- subroutine [readw_and_open](#) (file_or_extension, ierr, iopen)

6.1.1 Detailed Description

Definition at line 239 of file read_wout_mod.f.

6.1.2 Member Function/Subroutine Documentation

6.1.2.1 readw_and_open()

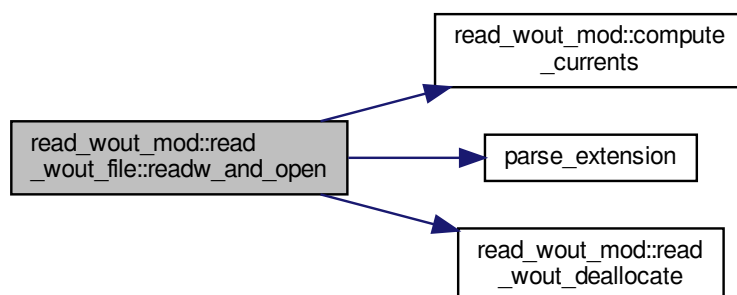
```
subroutine read_wout_mod::read_wout_file::readw_and_open (
    character(len=*), intent(in) file_or_extension,
    integer, intent(out) ierr,
    integer, optional iopen )
```

Definition at line 249 of file read_wout_mod.f.

References [read_wout_mod::ac](#), [read_wout_mod::ac_aux_f](#), [read_wout_mod::ac_aux_s](#), [read_wout_mod::ai](#), [read_wout_mod::ai_aux_f](#), [read_wout_mod::ai_aux_s](#), [read_wout_mod::am](#), [read_wout_mod::am_aux_f](#), [read_wout_mod::am_aux_s](#), [read_wout_mod::aminor](#), [read_wout_mod::aspect](#), [read_wout_mod::b0](#), [read_wout_mod::b0gradv](#), [read_wout_mod::beta_vol](#), [read_wout_mod::betapol](#), [read_wout_mod::betator](#), [read_wout_mod::betatot](#), [read_wout_mod::betaxis](#), [read_wout_mod::bmnc](#), [read_wout_mod::bmns](#), [read_wout_mod::bsubsmnc](#), [read_wout_mod::bsubsmns](#), [read_wout_mod::bsubumnc](#), [read_wout_mod::bsubumns](#), [read_wout_mod::bsubvmnc](#), [read_wout_mod::bsubvmns](#), [read_wout_mod::bsupumnc](#), [read_wout_mod::bsupumns](#), [read_wout_mod::bsupvmnc](#), [read_wout_mod::bsupvmns](#), [read_wout_mod::buc](#), [read_wout_mod::bvco](#), [read_wout_mod::chi](#), [read_wout_mod::chipf](#), [read_wout_mod::compute_currents\(\)](#), [mgrid_mod::curlabel](#), [read_wout_mod::dcurr](#), [read_wout_mod::dgeod](#), [read_wout_mod::dmerc](#), [read_wout_mod::dshear](#), [read_wout_mod::dwell](#), [read_wout_mod::equif](#), [read_wout_mod::extcur](#), [read_wout_mod::fsql](#), [read_wout_mod::fsqr](#), [read_wout_mod::fsqt](#),

read_wout_mod::fsqz, read_wout_mod::ftolv, read_wout_mod::gamma, read_wout_mod::gmnc, read_wout_mod::gmns, read_wout_mod::iasym, read_wout_mod::ierr_vmec, read_wout_mod::imse, read_wout_mod::input_extension, read_wout_mod::ionlarmor, read_wout_mod::iotaf, read_wout_mod::iotas, read_wout_mod::isigng, read_wout_mod::itfsq, read_wout_mod::itor, read_wout_mod::jcuru, read_wout_mod::jcurv, read_wout_mod::jdotb, read_wout_mod::lasym, read_wout_mod::lmnc, read_wout_mod::lmns, read_wout_mod::lthreed, read_wout_mod::lwout_opened, read_wout_mod::mass, read_wout_mod::mgrid_file, mgrid_mod::mgrid_mode, read_wout_mod::mnmax, read_wout_mod::mnmax_nyq, read_wout_mod::mnyq, read_wout_mod::mpol, stel_constants::mu0, mgrid_mod::nextcur, read_wout_mod::nfp, read_wout_mod::niter, read_wout_mod::nnyq, read_wout_mod::ns, read_wout_mod::ntor, read_wout_mod::overr, parse_extension(), read_wout_mod::pcurr_type, read_wout_mod::phi, read_wout_mod::phip, read_wout_mod::phipf, read_wout_mod::piota_type, read_wout_mod::pmass_type, read_wout_mod::potvac, read_wout_mod::pres, read_wout_mod::presf, read_wout_mod::qfact, read_wout_mod::raxis, read_wout_mod::rbtor, read_wout_mod::rbtor0, read_wout_mod::read_wout_deallocate(), read_wout_mod::rmajor, read_wout_mod::rmax_surf, read_wout_mod::rmin_surf, read_wout_mod::rmnc, read_wout_mod::rmns, read_wout_mod::specw, read_wout_mod::version_, read_wout_mod::vn_abeta, read_wout_mod::vn_ac, read_wout_mod::vn_ac_aux_f, read_wout_mod::vn_ac_aux_s, read_wout_mod::vn_actit, read_wout_mod::vn_ai, read_wout_mod::vn_ai_aux_f, read_wout_mod::vn_ai_aux_s, read_wout_mod::vn_am, read_wout_mod::vn_am_aux_f, read_wout_mod::vn_am_aux_s, read_wout_mod::vn_amin, read_wout_mod::vn_aspect, read_wout_mod::vn_asym, read_wout_mod::vn_b0, read_wout_mod::vn_beta, read_wout_mod::vn_betah, read_wout_mod::vn_bgrv, read_wout_mod::vn_bmnc, read_wout_mod::vn_bmns, read_wout_mod::vn_bsubsmnc, read_wout_mod::vn_bsubsmns, read_wout_mod::vn_bsubumnc, read_wout_mod::vn_bsubumns, read_wout_mod::vn_bsubvmnc, read_wout_mod::vn_bsubvmns, read_wout_mod::vn_bsupumnc, read_wout_mod::vn_bsupumns, read_wout_mod::vn_bsupvmnc, read_wout_mod::vn_bsupvmns, read_wout_mod::vn_buco, read_wout_mod::vn_bvco, read_wout_mod::vn_chi, read_wout_mod::vn_chipf, read_wout_mod::vn_ctor, read_wout_mod::vn_curlab, read_wout_mod::vn_equif, read_wout_mod::vn_error, read_wout_mod::vn_extcur, read_wout_mod::vn_extension, read_wout_mod::vn_fp, read_wout_mod::vn_free, read_wout_mod::vn_fsq, read_wout_mod::vn_fsql, read_wout_mod::vn_fsqr, read_wout_mod::vn_fsqz, read_wout_mod::vn_ftolv, read_wout_mod::vn_gam, read_wout_mod::vn_gmnc, read_wout_mod::vn_gmns, read_wout_mod::vn_iotaf, read_wout_mod::vn_iotah, read_wout_mod::vn_jcuru, read_wout_mod::vn_jcurv, read_wout_mod::vn_jdotb, read_wout_mod::vn_lar, read_wout_mod::vn_lmnc, read_wout_mod::vn_lmns, read_wout_mod::vn_magen, read_wout_mod::vn_mass, read_wout_mod::vn_maxit, read_wout_mod::vn_maxmod, read_wout_mod::vn_maxmod_nyq, read_wout_mod::vn_maxr, read_wout_mod::vn_maxz, read_wout_mod::vn_mcurr, read_wout_mod::vn_merc, read_wout_mod::vn_mgeo, mgrid_mod::vn_mgmode, read_wout_mod::vn_mgrid, read_wout_mod::vn_minr, read_wout_mod::vn_modb, read_wout_mod::vn_mshear, read_wout_mod::vn_mwell, mgrid_mod::vn_nextcur, read_wout_mod::vn_overr, read_wout_mod::vn_pbeta, read_wout_mod::vn_pcurr_type, read_wout_mod::vn_phi, read_wout_mod::vn_phip, read_wout_mod::vn_phipf, read_wout_mod::vn_piota_type, read_wout_mod::vn_pmass_type, read_wout_mod::vn_pmod, read_wout_mod::vn_pmod_nyq, read_wout_mod::vn_polmod, read_wout_mod::vn_potvac, read_wout_mod::vn_presf, read_wout_mod::vn_presh, read_wout_mod::vn_qfact, read_wout_mod::vn_racc, read_wout_mod::vn_racs, read_wout_mod::vn_radnod, read_wout_mod::vn_rbt0, read_wout_mod::vn_rbt1, read_wout_mod::vn_rmaj, read_wout_mod::vn_rmnc, read_wout_mod::vn_rmns, read_wout_mod::vn_sgs, read_wout_mod::vn_specw, read_wout_mod::vn_tbeta, read_wout_mod::vn_therm, read_wout_mod::vn_tmod, read_wout_mod::vn_tmod_nyq, read_wout_mod::vn_tormod, read_wout_mod::vn_version, read_wout_mod::vn_vol, read_wout_mod::vn_vp, read_wout_mod::vn_wdot, read_wout_mod::vn_zacc, read_wout_mod::vn_zacs, read_wout_mod::vn_zmnc, read_wout_mod::vn_zmns, read_wout_mod::volavgb, read_wout_mod::volume, read_wout_mod::vp, read_wout_mod::wb, read_wout_mod::wdot, read_wout_mod::wp, read_wout_mod::xm, read_wout_mod::xm_nyq, read_wout_mod::xn, read_wout_mod::xn_nyq, read_wout_mod::zaxis, read_wout_mod::zmax_surf, read_wout_mod::zmnc, and read_wout_mod::zmns.

Here is the call graph for this function:



Chapter 7

File Documentation

7.1 README.md File Reference

7.2 src/add_fluxes.f90 File Reference

Add the magnetic fluxes to the tangential derivatives of λ to arrive at the contravariant magnetic field components B^θ and B^ζ .

Functions/Subroutines

- subroutine [add_fluxes](#) (overg, bsupu, bsupv)

Add the magnetic fluxes to the tangential derivatives of λ to arrive at the contravariant magnetic field components B^θ and B^ζ .

7.2.1 Detailed Description

Add the magnetic fluxes to the tangential derivatives of λ to arrive at the contravariant magnetic field components B^θ and B^ζ .

7.2.2 Function/Subroutine Documentation

7.2.2.1 add_fluxes()

```
subroutine add_fluxes (  
    real(rprec), dimension(nrzt), intent(in) overg,  
    real(rprec), dimension(nrzt), intent(inout) bsupu,  
    real(rprec), dimension(nrzt), intent(inout) bsupv )
```

Add the magnetic fluxes to the tangential derivatives of λ to arrive at the contravariant magnetic field components B^θ and B^ζ .

Parameters

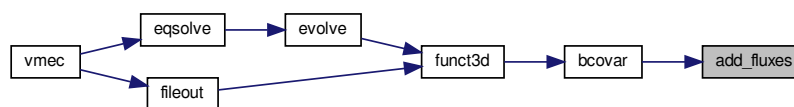
<i>overg</i>	$1/\sqrt{g}$
<i>bsupu</i>	B^θ
<i>bsupv</i>	B^ζ

Definition at line 11 of file `add_fluxes.f90`.

References `realspace::chip`, `vmec_main::chipf`, `vmec_main::chips`, `realspace::guu`, `realspace::guv`, `vmec_main::icurv`, `vmec_main::iotaf`, `vmec_main::iotas`, `vmec_main::phips`, and `realspace::wint`.

Referenced by `bcovar()`.

Here is the caller graph for this function:



7.3 src/alias.f90 File Reference

Fourier transform alias force and also return intermediate output.

Functions/Subroutines

- subroutine [alias](#) (`gcons`, `ztemp`, `gcs`, `gsc`, `gcc`, `gss`)
Fourier transform alias force from ztemp to gcons and also return intermediate output in g(c,s)(c,s)

7.3.1 Detailed Description

Fourier transform alias force and also return intermediate output.

7.3.2 Function/Subroutine Documentation

7.3.2.1 alias()

```

subroutine alias (
    real(rprec), dimension(ns*nzeta,ntheta3), intent(out) gcons,
    real(rprec), dimension(ns*nzeta,ntheta3), intent(in) ztemp,
    real(rprec), dimension(ns,0:ntor,0:mpoll), intent(inout) gcs,
    real(rprec), dimension(ns,0:ntor,0:mpoll), intent(inout) gsc,
    real(rprec), dimension(ns,0:ntor,0:mpoll), intent(inout) gcc,
    real(rprec), dimension(ns,0:ntor,0:mpoll), intent(inout) gss )

```

Fourier transform alias force from `ztemp` to `gcons` and also return intermediate output in `g(c,s)(c,s)`

Parameters

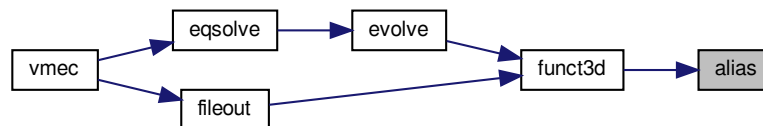
<i>gcons</i>	
<i>ztemp</i>	
<i>gcs</i>	
<i>gsc</i>	
<i>gcc</i>	
<i>gss</i>	

Definition at line 12 of file alias.f90.

References vmec_main::faccon, vmec_main::ireflect, and vmec_main::tcon.

Referenced by funct3d().

Here is the caller graph for this function:



7.4 src/allocate_func3d.f90 File Reference

allocate arrays required in [funct3d\(\)](#)

Functions/Subroutines

- subroutine [allocate_func3d](#)
allocate arrays required in [funct3d\(\)](#)

7.4.1 Detailed Description

allocate arrays required in [funct3d\(\)](#)

7.4.2 Function/Subroutine Documentation

7.4.2.1 allocate_funct3d()

```
subroutine allocate_funct3d
```

allocate arrays required in [funct3d\(\)](#)

Definition at line 6 of file `allocate_funct3d.f90`.

References `vforces::armn`, `vforces::armn_e`, `vforces::armn_o`, `vforces::azmn`, `vforces::azmn_e`, `vforces::azmn_o`, `vforces::blmn`, `vforces::blmn_e`, `vforces::blmn_o`, `vforces::brmn`, `vforces::brmn_e`, `vforces::brmn_o`, `vforces::bzmn`, `vforces::bzmn_e`, `vforces::bzmn_o`, `vforces::clmn`, `vforces::clmn_e`, `vforces::clmn_o`, `vforces::crmn`, `vforces::crmn_e`, `vforces::crmn_o`, `vforces::czmn`, `vforces::czmn_e`, `vforces::czmn_o`, `realspace::extra1`, `realspace::extra2`, `realspace::extra3`, `realspace::extra4`, `free_mem_funct3d()`, `realspace::gcon`, `realspace::guu`, `realspace::guv`, `realspace::gvv`, `realspace::r1`, `realspace::rcon`, `realspace::rcon0`, `realspace::ru`, `realspace::ru0`, `realspace::rv`, `realspace::z1`, `realspace::zcon`, `realspace::zcon0`, `realspace::zu`, `realspace::zu0`, and `realspace::zv`.

Referenced by `allocate_ns()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.5 src/allocate_ns.f90 File Reference

allocate arrays depending on the number of flux surfaces `ns`

Functions/Subroutines

- subroutine [allocate_ns](#) (`linterp`, `neqs_old`)
allocate arrays depending on the number of flux surfaces `ns`

7.5.1 Detailed Description

allocate arrays depending on the number of flux surfaces `ns`

7.5.2 Function/Subroutine Documentation

7.5.2.1 allocate_ns()

```
subroutine allocate_ns (
    logical, intent(in) linterp,
    integer, intent(in) neqs_old )
```

allocate arrays depending on the number of flux surfaces `ns`

Parameters

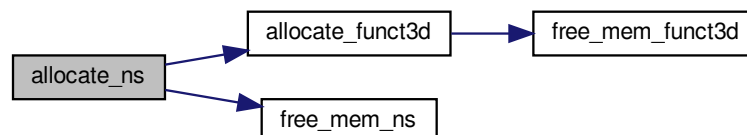
<i>linterp</i>	interpolate from coars to finer mesh?
<i>neqs_old</i>	previous number of degrees-of-freedom, i.e., Fourier coefficients for R , Z and λ

Definition at line 8 of file `allocate_ns.f90`.

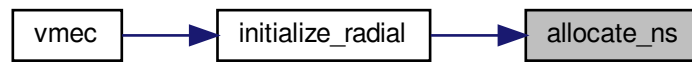
References `allocate_func3d()`, `vmec_main::ard`, `vmec_main::arm`, `vmec_main::azd`, `vmec_main::azm`, `vmec_main::bdamp`, `vmec_main::bdotb`, `vmec_main::bdotgradv`, `vmec_main::beta_vol`, `vmec_main::blam`, `vmec_main::bmax`, `vmec_main::bmin`, `vmec_main::brd`, `vmec_main::brm`, `vmec_main::bucf`, `vmec_main::bvco`, `vmec_main::bvcof`, `vmec_main::bzd`, `vmec_main::bzm`, `vmec_main::chi`, `realspace::chip`, `vmec_main::chipf`, `vmec_main::chips`, `vmec_main::clam`, `vmec_main::crd`, `vmec_main::dlam`, `vmec_main::equiv`, `vmec_main::faclam`, `fbal::frcc_fac`, `free_mem_ns()`, `fbal::fzsc_fac`, `xstuff::gc`, `vmec_main::icurv`, `vmec_main::iotaf`, `vmec_main::iotas`, `vmec_main::ireflect`, `vmec_main::jcuru`, `vmec_main::jcurv`, `vmec_main::jdotb`, `vmec_main::jpar2`, `vmec_main::jperp2`, `vmec_main::mass`, `vmec_main::neqs`, `vmec_params::ntmax`, `vmec_main::overr`, `vmec_main::phi`, `realspace::phip`, `vmec_main::phipf`, `vmec_main::phips`, `vmec_main::pres`, `vmec_main::presf`, `vmec_main::presgrad`, `vmec_main::psi`, `fbal::rru_fac`, `fbal::rzu_fac`, `xstuff::scalxc`, `realspace::shalf`, `vmec_main::sm`, `vmec_main::sp`, `vmec_main::specw`, `realspace::sqrts`, `vmec_main::tcon`, `vmec_main::vp`, `vmec_main::vpphi`, `realspace::wint`, `xstuff::xc`, `xstuff::xcdot`, `xstuff::xsave`, `xstuff::xstore`, `vmec_main::yellip`, `vmec_main::ygeo`, `vmec_main::yinden`, `vmec_main::yshift`, and `vmec_main::ytrian`.

Referenced by `initialize_radial()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.6 src/allocate_nunv.f90 File Reference

allocate arrays depending on the number of Fourier coefficients `nunv`

Functions/Subroutines

- subroutine [allocate_nunv](#)
allocate arrays depending on the number of Fourier coefficients `nunv`

7.6.1 Detailed Description

allocate arrays depending on the number of Fourier coefficients `nunv`

7.6.2 Function/Subroutine Documentation

7.6.2.1 allocate_nunv()

```
subroutine allocate_nunv
```

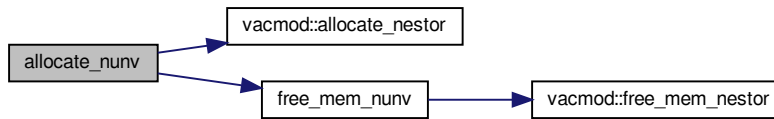
allocate arrays depending on the number of Fourier coefficients `nunv`

Definition at line 6 of file `allocate_nunv.f90`.

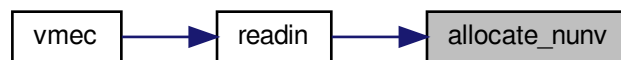
References `vacmod::allocate_nestor()`, `vmec_main::bsqsav`, `vmec_main::bsubu0`, `vmec_main::dbsq`, `free_mem_`↵
`nunv()`, `vmec_params::ntmax`, `vmec_main::rbsq`, `vmec_main::rmn_bdy`, and `vmec_main::zmn_bdy`.

Referenced by `readin()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.7 src/aspectratio.f90 File Reference

compute aspect-ratio (independent of elongation): $A = \langle R \rangle / \sqrt{\langle ab \rangle}$

Functions/Subroutines

- real(rprec) function [aspectratio](#) ()
compute aspect-ratio (independent of elongation): $A = \langle R \rangle / \sqrt{\langle ab \rangle}$ where $\pi \langle a \rangle^2 = \text{Area (toroidally averaged)}$ and $2\pi \langle R \rangle \text{Area} = \text{Volume}$

7.7.1 Detailed Description

compute aspect-ratio (independent of elongation): $A = \langle R \rangle / \sqrt{\langle ab \rangle}$

7.7.2 Function/Subroutine Documentation

7.7.2.1 [aspectratio\(\)](#)

real(rprec) function `aspectratio`

compute aspect-ratio (independent of elongation): $A = \langle R \rangle / \sqrt{\langle ab \rangle}$ where $\pi \langle a \rangle^2 = \text{Area (toroidally averaged)}$ and $2\pi \langle R \rangle \text{Area} = \text{Volume}$

Definition at line 7 of file `aspectratio.f90`.

References `vmec_io::aminor_p`, `vmec_io::cross_area_p`, `realspace::r1`, `vmec_io::rmajor_p`, `vmec_io::volume_p`, `realspace::wint`, and `realspace::zu`.

7.8 src/bcovar.f90 File Reference

Compute the covariant components of the magnetic field B_θ , B_ζ .

Functions/Subroutines

- subroutine `bcovar` (`lu`, `lv`)
Compute the covariant components of the magnetic field B_θ , B_ζ .

7.8.1 Detailed Description

Compute the covariant components of the magnetic field B_θ , B_ζ .

7.8.2 Function/Subroutine Documentation

7.8.2.1 `bcovar()`

```
subroutine bcovar (
    real(rprec), dimension(nrzt,0:1), intent(inout) lu,
    real(rprec), dimension(nrzt,0:1), intent(inout) lv )
```

Compute the covariant components of the magnetic field B_θ , B_ζ .

Parameters

<i>lu</i>	$\partial\lambda/\partial\theta$
<i>lv</i>	$\partial\lambda/\partial\zeta$

R12 from RP in force

Norm, unpreconditioned R,Z forces

Norm for preconditioned R,Z forces

Norm for unpreconditioned Lambda force

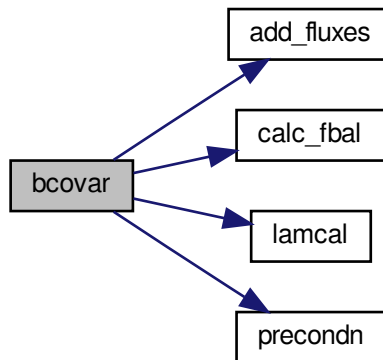
Definition at line 8 of file `bcovar.f90`.

References `add_fluxes()`, `vmec_main::ard`, `vmec_main::arm`, `vforces::armn_e`, `vforces::armn_o`, `vmec_main::azd`, `vmec_main::azm`, `vforces::azmn_e`, `vforces::azmn_o`, `vmec_main::bdamp`, `vforces::blmn_e`, `vforces::blmn_o`, `vmec_main::brd`, `vmec_main::brm`, `vforces::brmn_e`, `vforces::brmn_o`, `vmec_main::buc`, `vmec_main::bvco`, `vmec_main::bzd`, `vmec_main::bzm`, `vforces::bzmn_e`, `vforces::bzmn_o`, `calc_fbal()`, `vforces::clmn_e`, `vforces::clmn_o`, `vmec_main::crd`, `vmec_main::ctor`, `realspace::extra1`, `realspace::extra2`, `realspace::extra3`, `realspace::extra4`, `vmec_main::fnorm`, `vmec_main::fnorm1`, `vmec_main::fnorml`, `fbal::frcc_fac`, `fbal::fzsc_fac`, `realspace::guu`, `realspace::guv`, `realspace::gvv`, `vmec_main::hs`, `vmec_main::iequi`, `vmec_main::irzloff`, `vmec_main::iter1`,

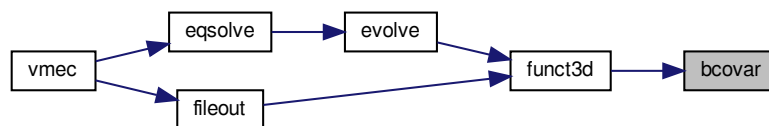
vmec_main::iter2, lamcal(), vmec_params::lamscale, vmec_main::lthreed, vmec_main::mass, vmec_params::ns4, vmec_params::pdamp, vmec_main::phipf, precondition(), vmec_main::pres, vmec_main::r0scale, realspace::r1, vmec_main::rbtor, vmec_main::rbtor0, fbal::rru_fac, realspace::ru, realspace::ru0, realspace::rv, fbal::rzu_fac, realspace::shalf, vmec_params::signgs, realspace::sqrts, vmec_main::tcon, vmec_main::voli, vmec_main::vp, vmec_main::wb, realspace::wint, vmec_main::wp, xstuff::xc, realspace::z1, realspace::zu, realspace::zu0, and realspace::zv.

Referenced by funct3d().

Here is the call graph for this function:



Here is the caller graph for this function:



7.9 src/bextrema.f90 File Reference

Computes minimum and maximum $|\mathbf{B}|$ along ζ between two angle lines ($\theta = 0, \pi$).

Functions/Subroutines

- subroutine `bextrema` (modb, bmin, bmax, nzeta, ntheta)

Computes minimum and maximum $|\mathbf{B}|$ along ζ between two angle lines ($\theta = 0, \pi$).

7.9.1 Detailed Description

Computes minimum and maximum $|\mathbf{B}|$ along ζ between two angle lines ($\theta = 0, \pi$).

7.9.2 Function/Subroutine Documentation

7.9.2.1 bextrema()

```
subroutine bextrema (
    real(rprec), dimension(nzeta,ntheta), intent(in) modb,
    real(rprec), dimension(ntheta), intent(out) bmin,
    real(rprec), dimension(ntheta), intent(out) bmax,
    integer, intent(in) nzeta,
    integer, intent(in) ntheta )
```

Computes minimum and maximum $|\mathbf{B}|$ along ζ between two angle lines ($\theta = 0, \pi$).

Parameters

<i>modb</i>	magnitude of magnetic field $ \mathbf{B} $
<i>bmin</i>	minimum value of $ \mathbf{B} $
<i>bmax</i>	maximum value of $ \mathbf{B} $
<i>nzeta</i>	number of grid points in toroidal direction
<i>ntheta</i>	number of grid points in poloidal direction

Definition at line 11 of file bextrema.f90.

Referenced by eqfor().

Here is the caller graph for this function:



7.10 src/bss.f90 File Reference

Computes br, bphi, bz, bsubs on half-radial mesh.

Functions/Subroutines

- subroutine [bss](#) (r12, rs, zs, ru12, zu12, bsubs, bsupu, bsupv, br, bphi, bz)
Computes br, bphi, bz, bsubs on half-radial mesh.

7.10.1 Detailed Description

Computes br, bphi, bz, bsubs on half-radial mesh.

7.10.2 Function/Subroutine Documentation

7.10.2.1 [bss\(\)](#)

```
subroutine bss (
    real(rprec), dimension(nrzt), intent(in) r12,
    real(rprec), dimension(nrzt), intent(in) rs,
    real(rprec), dimension(nrzt), intent(in) zs,
    real(rprec), dimension(nrzt), intent(in) ru12,
    real(rprec), dimension(nrzt), intent(in) zu12,
    real(rprec), dimension(nrzt), intent(out) bsubs,
    real(rprec), dimension(nrzt), intent(in) bsupu,
    real(rprec), dimension(nrzt), intent(in) bsupv,
    real(rprec), dimension(nrzt), intent(out) br,
    real(rprec), dimension(nrzt), intent(out) bphi,
    real(rprec), dimension(nrzt), intent(out) bz )
```

Computes br, bphi, bz, bsubs on half-radial mesh.

Parameters

<i>r12</i>	R^2
<i>rs</i>	$\partial R / \partial s$
<i>zs</i>	$\partial Z / \partial s$
<i>ru12</i>	$(\partial R / \partial \theta)^2$
<i>zu12</i>	$(\partial Z / \partial \theta)^2$
<i>bsubs</i>	covariant component of magnetic field B_s
<i>bsupu</i>	contravariant component of magnetic field B^θ
<i>bsupv</i>	contravariant component of magnetic field B^ζ
<i>br</i>	cylindrical component of magnetic field B^R
<i>bphi</i>	cylindrical component of magnetic field B^φ
<i>bz</i>	cylindrical component of magnetic field B^Z

Definition at line 17 of file bss.f90.

References [realspace::r1](#), [realspace::rv](#), [realspace::shalf](#), [realspace::z1](#), and [realspace::zv](#).

Referenced by eqfor().

Here is the caller graph for this function:



7.11 src/calc_fbal.f90 File Reference

Compute flux-surface averaged radial force balance $\nabla p - \langle \mathbf{j} \times \mathbf{B} \rangle$.

Functions/Subroutines

- subroutine [calc_fbal](#) (bsubu, bsubv)
Compute flux-surface averaged radial force balance $\nabla p - \langle \mathbf{j} \times \mathbf{B} \rangle$.

7.11.1 Detailed Description

Compute flux-surface averaged radial force balance $\nabla p - \langle \mathbf{j} \times \mathbf{B} \rangle$.

7.11.2 Function/Subroutine Documentation

7.11.2.1 calc_fbal()

```

subroutine calc_fbal (
    real(dp), dimension(1:nrzt), intent(in) bsubu,
    real(dp), dimension(1:nrzt), intent(in) bsubv )

```

Compute flux-surface averaged radial force balance $\nabla p - \langle \mathbf{j} \times \mathbf{B} \rangle$.

Parameters

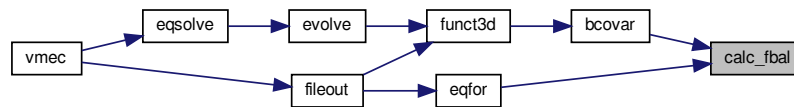
<i>bsubu</i>	covariant component of magnetic field B_θ
<i>bsubv</i>	covariant component of magnetic field B_ζ

Definition at line 8 of file calc_fbal.f90.

References vmec_main::bucor, vmec_main::bvco, vmec_main::chipf, stel_kinds::dp, vmec_main::equiv, vmec_main::jcuru, vmec_main::jcurv, vmec_dim::nrzt, vmec_dim::ns, vmec_dim::ns1, vmec_main::ohs, vmec_main::phipf, vmec_main::pres, vmec_main::presgrad, vmec_params::signgs, vmec_main::vp, vmec_main::vpphi, and realspace::wint.

Referenced by bcovar(), and eqfor().

Here is the caller graph for this function:



7.12 src/CMakeLists.txt File Reference

Functions

- [add_subdirectory](#) (data) add_subdirectory(ezcdf) add_subdirectory(NESTOR) [list](#)(APPEND vmec_sources \$

7.12.1 Function Documentation

7.12.1.1 add_subdirectory()

```
add_subdirectory (
    data )
```

Definition at line 2 of file CMakeLists.txt.

7.13 src/data/CMakeLists.txt File Reference

Functions

- [list](#) (APPEND vmec_sources \${CMAKE_CURRENT_SOURCE_DIR}/realspace.f90 \${CMAKE_CURRENT_SOURCE_DIR}/stel_constants.f90 \${CMAKE_CURRENT_SOURCE_DIR}/stel_kinds.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vforces.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vmec_dim.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vmec_input.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vmec_io.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vmec_main.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vmec_params.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vmec_persistent.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vmecier.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vparams.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vsvd0.f90 \${CMAKE_CURRENT_SOURCE_DIR}/xstuff.f90 \${CMAKE_CURRENT_SOURCE_DIR}/fbal.f90) set(vmec_sources "\$

7.13.1 Function Documentation

7.13.1.1 list()

```
list (
    APPEND vmec_sources ${CMAKE_CURRENT_SOURCE_DIR}/realspace.f90 ${CMAKE_CURRENT_↵
_SOURCE_DIR}/stel_constants.f90 ${CMAKE_CURRENT_SOURCE_DIR}/stel_kinds.f90 ${CMAKE_CURRENT_↵
_SOURCE_DIR}/vforces.f90 ${CMAKE_CURRENT_SOURCE_DIR}/vmec_dim.f90 ${CMAKE_CURRENT_SOURCE_↵
_DIR}/vmec_input.f90 ${CMAKE_CURRENT_SOURCE_DIR}/vmec_io.f90 ${CMAKE_CURRENT_SOURCE_DIR}/vmec_↵
_main.f90 ${CMAKE_CURRENT_SOURCE_DIR}/vmec_params.f90 ${CMAKE_CURRENT_SOURCE_DIR}/vmec_↵
persistent.f90 ${CMAKE_CURRENT_SOURCE_DIR}/vmecier.f90 ${CMAKE_CURRENT_SOURCE_DIR}/vparams.↵
f90 ${CMAKE_CURRENT_SOURCE_DIR}/vsvd0.f90 ${CMAKE_CURRENT_SOURCE_DIR}/xstuff.f90 ${CMAKE_↵
CURRENT_SOURCE_DIR}/fbal.  f90 )
```

Definition at line 2 of file CMakeLists.txt.

7.14 src/NESTOR/CMakeLists.txt File Reference

Functions

- [add_subdirectory](#) (data) [list](#)(APPEND vmec_sources \$

7.14.1 Function Documentation

7.14.1.1 add_subdirectory()

```
add_subdirectory (
    data )
```

Definition at line 2 of file CMakeLists.txt.

7.15 src/NESTOR/data/CMakeLists.txt File Reference

Functions

- [list](#) (APPEND vmec_sources \${CMAKE_CURRENT_SOURCE_DIR}/vacmod.f90 \${CMAKE_CURRENT_↵
_SOURCE_DIR}/vacmod0.f90 \${CMAKE_CURRENT_SOURCE_DIR}/vac_persistent.f90 \${CMAKE_↵
CURRENT_SOURCE_DIR}/nestor_io.f90) set(vmec_sources "\$

7.15.1 Function Documentation

7.15.1.1 list()

```
list (
    APPEND vmec_sources ${CMAKE_CURRENT_SOURCE_DIR}/vacmod.f90 ${CMAKE_CURRENT_↵
SOURCE_DIR}/vacmod0.f90 ${CMAKE_CURRENT_SOURCE_DIR}/vac_persistent.f90 ${CMAKE_CURRENT_↵
SOURCE_DIR}/nestor_io. f90 )
```

Definition at line 2 of file CMakeLists.txt.

7.16 src/convert.f90 File Reference

Convert internal mode representation to standard form for output (coefficients of $\cos(\mu\text{-}nv)$, $\sin(\mu\text{-}nv)$ without internal `mscale`, `nscale` norms).

Functions/Subroutines

- subroutine [convert](#) (`rmnc`, `zmns`, `lmns`, `rmns`, `zmnc`, `lmnc`, `rzl_array`, `js`)

Convert internal mode representation to standard form for output (coefficients of $\cos(\mu\text{-}nv)$, $\sin(\mu\text{-}nv)$ without internal `mscale`, `nscale` norms).

7.16.1 Detailed Description

Convert internal mode representation to standard form for output (coefficients of $\cos(\mu\text{-}nv)$, $\sin(\mu\text{-}nv)$ without internal `mscale`, `nscale` norms).

7.16.2 Function/Subroutine Documentation

7.16.2.1 convert()

```
subroutine convert (
    real(rprec), dimension(mnmax), intent(out) rmnc,
    real(rprec), dimension(mnmax), intent(out) zmns,
    real(rprec), dimension(mnmax), intent(out) lmns,
    real(rprec), dimension(mnmax), intent(out) rmns,
    real(rprec), dimension(mnmax), intent(out) zmnc,
    real(rprec), dimension(mnmax), intent(out) lmnc,
    real(rprec), dimension(ns, 0:ntor, 0:mpoll, 3*ntmax), intent(in) rzl_array,
    integer, intent(in) js )
```

Convert internal mode representation to standard form for output (coefficients of $\cos(\mu\text{-}nv)$, $\sin(\mu\text{-}nv)$ without internal `mscale`, `nscale` norms).

Parameters

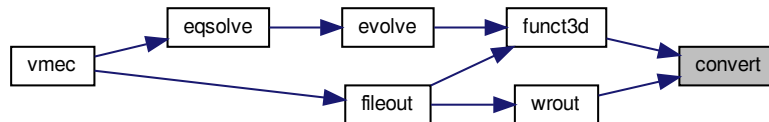
<i>rmnc</i>	stellarator-symmetric Fourier coefficients of R
<i>zmns</i>	stellarator-symmetric Fourier coefficients of Z
<i>lmns</i>	stellarator-symmetric Fourier coefficients of λ
<i>rmns</i>	non-stellarator-symmetric Fourier coefficients of R
<i>zmnc</i>	non-stellarator-symmetric Fourier coefficients of Z
<i>lmnc</i>	non-stellarator-symmetric Fourier coefficients of λ

Definition at line 16 of file convert.f90.

References vmec_main::lthreed, vmec_params::mscale, vmec_params::nscale, vmec_params::ntmax, vmec_params::rcc, vmec_params::rcs, vmec_params::rsc, vmec_params::rss, vmec_params::zcc, vmec_params::zcs, vmec_params::zsc, and vmec_params::zss.

Referenced by funct3d(), and wrout().

Here is the caller graph for this function:



7.17 src/data/fbal.f90 File Reference

Modules

- module [fbal](#)

Variables

- real(dp), dimension(:), allocatable [fbal::rzu_fac](#)
- real(dp), dimension(:), allocatable [fbal::rru_fac](#)
- real(dp), dimension(:), allocatable [fbal::frcc_fac](#)
- real(dp), dimension(:), allocatable [fbal::fzsc_fac](#)

7.18 src/data/realspace.f90 File Reference

Modules

- module [realspace](#)

Variables

- real(rprec), dimension(:,:), allocatable `realspace::r1`
- real(rprec), dimension(:,:), allocatable `realspace::ru`
- real(rprec), dimension(:,:), allocatable `realspace::rv`
- real(rprec), dimension(:,:), allocatable, target `realspace::z1`
- real(rprec), dimension(:,:), allocatable `realspace::zu`
- real(rprec), dimension(:,:), allocatable `realspace::zv`
- real(rprec), dimension(:,:), allocatable `realspace::rcon`
- real(rprec), dimension(:,:), allocatable `realspace::zcon`
- real(rprec), dimension(:), allocatable `realspace::guu`
- real(rprec), dimension(:), allocatable `realspace::guv`
- real(rprec), dimension(:), allocatable `realspace::gvv`
- real(rprec), dimension(:), allocatable `realspace::ru0`
- real(rprec), dimension(:), allocatable `realspace::zu0`
- real(rprec), dimension(:), allocatable `realspace::gcon`
- real(rprec), dimension(:), allocatable `realspace::rcon0`
- real(rprec), dimension(:), allocatable `realspace::zcon0`
- real(rprec), dimension(:), allocatable `realspace::phip`
radial derivative of $\phi/(2\pi)$ on half-grid
- real(rprec), dimension(:), allocatable `realspace::chip`
radial derivative of $\chi/(2\pi)$ on half-grid
- real(rprec), dimension(:), allocatable `realspace::shalf`
 \sqrt{s} , two-dimensional array on half-grid
- real(rprec), dimension(:), allocatable `realspace::sqrts`
 \sqrt{s} , two-dimensional array on full-grid
- real(rprec), dimension(:), allocatable `realspace::wint`
two-dimensional array for normalizing angle integrations
- real(rprec), dimension(:,:), allocatable, target `realspace::extra1`
- real(rprec), dimension(:,:), allocatable, target `realspace::extra2`
- real(rprec), dimension(:,:), allocatable, target `realspace::extra3`
- real(rprec), dimension(:,:), allocatable, target `realspace::extra4`

7.19 src/data/stel_constants.f90 File Reference

Modules

- module `stel_constants`

Variables

- real(dp), parameter `stel_constants::pi` = 3.14159265358979323846264338328_dp
- real(dp), parameter `stel_constants::pio2` = pi/2
- real(dp), parameter `stel_constants::twopi` = 2*pi
- real(dp), parameter `stel_constants::sqrt2` = 1.41421356237309504880168872_dp
- real(dp), parameter `stel_constants::degree` = twopi / 360
- real(dp), parameter `stel_constants::one` = 1
- real(dp), parameter `stel_constants::zero` = 0
- real(dp), parameter `stel_constants::mu0` = 2 * twopi * 1.0e-7_dp

7.20 src/data/stel_kinds.f90 File Reference

Modules

- module [stel_kinds](#)

Variables

- integer, parameter [stel_kinds::rprec](#) = SELECTED_REAL_KIND(12, 100)
- integer, parameter [stel_kinds::iprec](#) = SELECTED_INT_KIND(8)
- integer, parameter [stel_kinds::cprec](#) = KIND((1.0_rprec, 1.0_rprec))
- integer, parameter [stel_kinds::dp](#) = rprec

7.21 src/data/vforces.f90 File Reference

Modules

- module [vforces](#)

Variables

- real(rprec), dimension(:), allocatable, target [vforces::armn](#)
- real(rprec), dimension(:), allocatable, target [vforces::azmn](#)
- real(rprec), dimension(:), allocatable, target [vforces::brmn](#)
- real(rprec), dimension(:), allocatable, target [vforces::bzm](#)
- real(rprec), dimension(:), allocatable, target [vforces::blmn](#)
- real(rprec), dimension(:), allocatable, target [vforces::crmn](#)
- real(rprec), dimension(:), allocatable, target [vforces::czmn](#)
- real(rprec), dimension(:), allocatable, target [vforces::clmn](#)
- real(rprec), dimension(:), pointer [vforces::armn_e](#)
- real(rprec), dimension(:), pointer [vforces::armn_o](#)
- real(rprec), dimension(:), pointer [vforces::azmn_e](#)
- real(rprec), dimension(:), pointer [vforces::azmn_o](#)
- real(rprec), dimension(:), pointer [vforces::brmn_e](#)
- real(rprec), dimension(:), pointer [vforces::brmn_o](#)
- real(rprec), dimension(:), pointer [vforces::bzm_e](#)
- real(rprec), dimension(:), pointer [vforces::bzm_o](#)
- real(rprec), dimension(:), pointer [vforces::blmn_e](#)
- real(rprec), dimension(:), pointer [vforces::blmn_o](#)
- real(rprec), dimension(:), pointer [vforces::crmn_e](#)
- real(rprec), dimension(:), pointer [vforces::crmn_o](#)
- real(rprec), dimension(:), pointer [vforces::czmn_e](#)
- real(rprec), dimension(:), pointer [vforces::czmn_o](#)
- real(rprec), dimension(:), pointer [vforces::clmn_e](#)
- real(rprec), dimension(:), pointer [vforces::clmn_o](#)

7.22 src/data/vmec_dim.f90 File Reference

Modules

- module [vmec_dim](#)

Variables

- integer [vmec_dim::mpol1](#)
- integer [vmec_dim::ntor1](#)
- integer [vmec_dim::mnmax](#)
- integer [vmec_dim::ntheta1](#)
- integer [vmec_dim::ntheta2](#)
- integer [vmec_dim::ntheta3](#)
- integer [vmec_dim::nznt](#)
- integer [vmec_dim::nrzt](#)
- integer [vmec_dim::mns](#)
- integer [vmec_dim::mnsiz](#)
- integer [vmec_dim::mnmax_nyq](#)
- integer [vmec_dim::ns](#)
- integer [vmec_dim::ns1](#)
- integer [vmec_dim::ns_maxval](#)

7.23 src/data/vmec_input.f90 File Reference

Modules

- module [vmec_input](#)

Functions/Subroutines

- subroutine [vmec_input::read_indata_namelist](#) (iunit, istat)
- subroutine [vmec_input::write_indata_namelist](#) (iunit, istat)

Variables

- integer, parameter [vmec_input::mpol_default](#) = 6
- integer, parameter [vmec_input::ntor_default](#) = 0
- integer, parameter [vmec_input::ns_default](#) = 31
- integer, parameter [vmec_input::niter_default](#) = 100
- real(rprec), parameter [vmec_input::ftol_default](#) = 1.E-10_dp
- integer [vmec_input::nfp](#)
- integer [vmec_input::ncurr](#)
- integer [vmec_input::nstep](#)
- integer [vmec_input::nvacskip](#)
- integer [vmec_input::mpol](#)
- integer [vmec_input::ntor](#)
- integer [vmec_input::ntheta](#)
- integer [vmec_input::nzeta](#)

- integer `vmec_input::mfilter_fbdy`
- integer `vmec_input::nfilter_fbdy`
- integer, dimension(100) `vmec_input::ns_array`
- integer, dimension(100) `vmec_input::niter_array`
- real(rprec), dimension(100) `vmec_input::ftol_array`
- real(rprec), dimension(-ntord:ntord, 0:mpol1d) `vmec_input::rbc`
- real(rprec), dimension(-ntord:ntord, 0:mpol1d) `vmec_input::zbs`
- real(rprec), dimension(-ntord:ntord, 0:mpol1d) `vmec_input::rbs`
- real(rprec), dimension(-ntord:ntord, 0:mpol1d) `vmec_input::zbc`
- real(rprec) `vmec_input::curtor`
- real(rprec) `vmec_input::delt`
- real(rprec) `vmec_input::tcon0`
- real(rprec) `vmec_input::gamma`
- real(rprec) `vmec_input::bloat`
- real(rprec) `vmec_input::pres_scale`
- real(rprec) `vmec_input::spres_ped`
value of s beyond which pressure profile is flat (pedestal)
- real(rprec) `vmec_input::phiedge`
value of real toroidal flux at plasma edge (s=1)
- real(rprec), dimension(0:20) `vmec_input::am`
*array of coefficients in phi-series for mass (NWT/m**2)*
- real(rprec), dimension(0:20) `vmec_input::ai`
array of coefficients in phi-series for iota (ncurr=0)
- real(rprec), dimension(0:20) `vmec_input::ac`
*array of coefficients in phi-series for the quantity d(lcurv)/ds = toroidal current density * Vprime, so lcurv(s) = ltor(s)
(used for ncurr=1)*
- real(rprec), dimension(1:20) `vmec_input::aphi`
- character(len=20) `vmec_input::pcurr_type`
- character(len=20) `vmec_input::piota_type`
- character(len=20) `vmec_input::pmass_type`
- real(rprec), dimension(ndatafmax) `vmec_input::am_aux_s`
- real(rprec), dimension(ndatafmax) `vmec_input::am_aux_f`
- real(rprec), dimension(ndatafmax) `vmec_input::ai_aux_s`
- real(rprec), dimension(ndatafmax) `vmec_input::ai_aux_f`
- real(rprec), dimension(ndatafmax) `vmec_input::ac_aux_s`
- real(rprec), dimension(ndatafmax) `vmec_input::ac_aux_f`
- real(rprec), dimension(0:ntord) `vmec_input::raxis_cc`
- real(rprec), dimension(0:ntord) `vmec_input::raxis_cs`
- real(rprec), dimension(0:ntord) `vmec_input::zaxis_cc`
- real(rprec), dimension(0:ntord) `vmec_input::zaxis_cs`
- real(rprec), dimension(nigroup) `vmec_input::extcur`
- logical `vmec_input::lfreeb`
- logical `vmec_input::lasym`
- logical `vmec_input::lbsubs`
- character(len=200) `vmec_input::mgrid_file`
- character(len=100) `vmec_input::input_extension`

7.24 src/data/vmec_io.f90 File Reference

Modules

- module `vmec_io`

Variables

- real(rprec) [vmec_io::volavgb](#)
- real(rprec) [vmec_io::ionlarmor](#)
- real(rprec) [vmec_io::aminor_p](#)
- real(rprec) [vmec_io::rmajor_p](#)
- real(rprec) [vmec_io::betatot](#)
- real(rprec) [vmec_io::betapol](#)
- real(rprec) [vmec_io::betator](#)
- real(rprec) [vmec_io::betaxis](#)
- real(rprec) [vmec_io::b0](#)
- real(rprec) [vmec_io::volume_p](#)
- real(rprec) [vmec_io::cross_area_p](#)
- real(rprec) [vmec_io::surf_area_p](#)
- real(rprec) [vmec_io::circum_p](#)
- real(rprec) [vmec_io::kappa_p](#)
- real(rprec) [vmec_io::rmax_surf](#)
- real(rprec) [vmec_io::rmin_surf](#)
- real(rprec) [vmec_io::zmax_surf](#)

7.25 src/data/vmec_main.f90 File Reference

Modules

- module [vmec_main](#)

Variables

- real(rprec), dimension(:,:), allocatable [vmec_main::ard](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::arm](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::brd](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::brm](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::azd](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::azm](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::bzd](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::bzm](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::bmin](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::bmax](#)
- real(rprec), dimension(:), allocatable [vmec_main::crd](#)
- real(rprec), dimension(:), allocatable [vmec_main::iotaf](#)
- real(rprec), dimension(:), allocatable [vmec_main::phipf](#)
- real(rprec), dimension(:), allocatable [vmec_main::chipf](#)
- real(rprec), dimension(:), allocatable [vmec_main::phi](#)
- real(rprec), dimension(:), allocatable [vmec_main::beta_vol](#)
- real(rprec), dimension(:), allocatable [vmec_main::jcuru](#)
- real(rprec), dimension(:), allocatable [vmec_main::jcurv](#)
- real(rprec), dimension(:), allocatable [vmec_main::jdotb](#)
- real(rprec), dimension(:), allocatable [vmec_main::buco](#)
- real(rprec), dimension(:), allocatable [vmec_main::bvco](#)
- real(rprec), dimension(:), allocatable [vmec_main::bdotgradv](#)
- real(rprec), dimension(:), allocatable [vmec_main::equiv](#)

- real(rprec), dimension(:), allocatable [vmec_main::specw](#)
- real(rprec), dimension(:), allocatable [vmec_main::tcon](#)
- real(rprec), dimension(:), allocatable [vmec_main::psi](#)
- real(rprec), dimension(:), allocatable [vmec_main::yellip](#)
- real(rprec), dimension(:), allocatable [vmec_main::yinden](#)
- real(rprec), dimension(:), allocatable [vmec_main::ytrian](#)
- real(rprec), dimension(:), allocatable [vmec_main::yshift](#)
- real(rprec), dimension(:), allocatable [vmec_main::ygeo](#)
- real(rprec), dimension(:), allocatable [vmec_main::overr](#)
- real(rprec), dimension(:), allocatable [vmec_main::sm](#)
- real(rprec), dimension(:), allocatable [vmec_main::sp](#)
- real(rprec), dimension(:), allocatable [vmec_main::pres](#)
- real(rprec), dimension(:), allocatable [vmec_main::vp](#)
- real(rprec), dimension(:), allocatable [vmec_main::jpar2](#)
- real(rprec), dimension(:), allocatable [vmec_main::jperp2](#)
- real(rprec), dimension(:), allocatable [vmec_main::bdotb](#)
- real(rprec), dimension(:), allocatable [vmec_main::blam](#)
- real(rprec), dimension(:), allocatable [vmec_main::clam](#)
- real(rprec), dimension(:), allocatable [vmec_main::dlam](#)
- real(rprec), dimension(:), allocatable [vmec_main::vpphi](#)
- real(rprec), dimension(:), allocatable [vmec_main::presgrad](#)
- real(rprec), dimension(:), allocatable [vmec_main::bdamp](#)
- real(rprec), dimension(:), allocatable [vmec_main::bucof](#)
- real(rprec), dimension(:), allocatable [vmec_main::bvcof](#)
- real(rprec), dimension(:), allocatable [vmec_main::chi](#)
- real(rprec), dimension(:), allocatable [vmec_main::presf](#)
*pressure profile on full-grid, mass/phi**gamma*
- real(rprec), dimension(:), allocatable [vmec_main::chips](#)
poloidal flux (same as chip), one-dimensional array
- real(rprec), dimension(:), allocatable [vmec_main::phips](#)
toroidal flux (same as phi), one-dimensional array
- real(rprec), dimension(:), allocatable [vmec_main::iotas](#)
rotational transform , on half radial mesh
- real(rprec), dimension(:), allocatable [vmec_main::icurv](#)
(-)toroidal current inside flux surface (vanishes like s)
- real(rprec), dimension(:), allocatable [vmec_main::mass](#)
mass profile on half-grid
- real(rprec), dimension(:,:,:), allocatable [vmec_main::faclam](#)
- real(rprec), dimension(:,:,:), allocatable [vmec_main::faclam0](#)
- real(rprec), dimension(:,:), allocatable [vmec_main::bsqsav](#)
- real(rprec), dimension(:), allocatable [vmec_main::bledge](#)
- real(rprec), dimension(:), allocatable [vmec_main::bpedge](#)
- real(rprec), dimension(:), allocatable [vmec_main::bzedge](#)
- real(rprec), dimension(:), allocatable [vmec_main::xcl0](#)
- real(rprec), dimension(0:mpol1d, 3) [vmec_main::xmpq](#)
- real(rprec), dimension(0:mpol1d) [vmec_main::faccon](#)
- real(rprec) [vmec_main::hs](#)
radial mesh size increment
- real(rprec) [vmec_main::currv](#)
- real(rprec) [vmec_main::aspect](#)
- real(rprec) [vmec_main::ohs](#)
- real(rprec) [vmec_main::voli](#)
- real(rprec) [vmec_main::r00](#)

- real(rprec) vmec_main::r0scale
- real(rprec) vmec_main::z00
- real(rprec) vmec_main::fsqsum0
- real(rprec) vmec_main::fnorm
- real(rprec) vmec_main::fsqr = 1
- real(rprec) vmec_main::fsqz = 1
- real(rprec) vmec_main::fsql = 1
- real(rprec) vmec_main::fnorm1
- real(rprec) vmec_main::fnorml
- real(rprec) vmec_main::fsqr1
- real(rprec) vmec_main::fsqz1
- real(rprec) vmec_main::fsql1
- real(rprec) vmec_main::fsq
- real(rprec) vmec_main::fedge
- real(rprec) vmec_main::wb
- real(rprec) vmec_main::wp
- real(rprec) vmec_main::router
- real(rprec) vmec_main::rinner
- real(rprec) vmec_main::ftolv
- real(rprec) vmec_main::otav
- time-step algorithm*
- real(rprec), dimension(ndamp) vmec_main::otau
- real(rprec), dimension(:, :, :), allocatable, target vmec_main::rmn_bdy
- real(rprec), dimension(:, :, :), allocatable, target vmec_main::zmn_bdy
- real(rprec), dimension(:), allocatable vmec_main::bsubu0
- real(rprec), dimension(:), allocatable vmec_main::dbsq
- real(rprec), dimension(:), allocatable vmec_main::rbsq
- real(rprec) vmec_main::rbtor
- real(rprec) vmec_main::rbtor0
- real(rprec) vmec_main::ctor
- real(rprec) vmec_main::delbsq
- real(rprec) vmec_main::res0
- real(rprec) vmec_main::delt0r
- real(rprec), dimension(ndatafmax) vmec_main::spfa
- real(rprec), dimension(ndatafmax) vmec_main::spfa2
- real(rprec), dimension(ndatafmax) vmec_main::hp
- real(rprec), dimension(ndatafmax) vmec_main::sifa
- real(rprec), dimension(ndatafmax) vmec_main::sifa2
- real(rprec), dimension(ndatafmax) vmec_main::hi
- logical vmec_main::lthree
- logical vmec_main::lconm1
- logical vmec_main::lflip
- from init_geometry*
- integer, dimension(:), allocatable vmec_main::ireflect
- two-dimensional array for computing 2pi-v angle*
- integer vmec_main::multi_ns_grid
- integer vmec_main::itfsq
- integer vmec_main::ndatap
- integer vmec_main::ndatai
- integer vmec_main::niterv
- max iterations for current multi-grid iteration*
- integer vmec_main::neqs
- total number of equations to evolve (size of xc)*
- integer vmec_main::irzloff

- *offset in xc array between R,Z,L components*
- integer `vmec_main::iequi`
counter used to call -EQFOR- at end of run
- integer `vmec_main::ijacob`
counter for number of times jacobian changes sign
- integer `vmec_main::irst`
"counter" monitoring sign of jacobian; resets R, Z, and Lambda when jacobian changes sign and decreases time step
- integer `vmec_main::iter1`
number of iterations at which the currently active evolution was branched off from
- integer `vmec_main::iter2`
total number of iterations
- integer `vmec_main::ivac`
counts number of free-boundary iterations
- integer `vmec_main::vacuum_calls` = 0

7.26 src/data/vmec_params.f90 File Reference

Modules

- module `vmec_params`

Variables

- integer, parameter `vmec_params::meven` = 0
parity selection label for even poloidal modes of R and Z
- integer, parameter `vmec_params::modd` = 1
parity selection label for odd poloidal modes of R and Z
- integer, parameter `vmec_params::ndamp` = 10
number of iterations over which damping is averaged
- integer, parameter `vmec_params::ns4` = 25
- integer, dimension(0:mpold), parameter `vmec_params::jmin1` = (/ 1,1,(2,ink=2,mpold) /)
starting js(m) values where R,Z are non-zero
- integer, dimension(0:mpold), parameter `vmec_params::jmin2` = (/ 1,2,(2,ink=2,mpold) /)
starting js(m) values for which R,Z are evolved
- integer, dimension(0:mpold), parameter `vmec_params::jlam` = (/ 2,2,(2,ink=2,mpold) /)
starting js(m) values for which Lambda is evolved
- integer, parameter `vmec_params::norm_term_flag` = 0
- integer, parameter `vmec_params::bad_jacobian_flag` = 1
- integer, parameter `vmec_params::jac75_flag` = 4
- integer, parameter `vmec_params::input_error_flag` = 5
- integer, parameter `vmec_params::phiedge_error_flag` = 7
- integer, parameter `vmec_params::ns_error_flag` = 8
- integer, parameter `vmec_params::misc_error_flag` = 9
- integer, parameter `vmec_params::successful_term_flag` = 11
- integer, parameter `vmec_params::restart_flag` = 1
- integer, parameter `vmec_params::readin_flag` = 2
- integer, parameter `vmec_params::timestep_flag` = 4
- integer, parameter `vmec_params::output_flag` = 8
- integer, parameter `vmec_params::cleanup_flag` = 16

- integer, parameter `vmec_params::reset_jacdt_flag` = 32
- real(rprec), parameter `vmec_params::pdamp` = 0.05_dp
- character(len= *), parameter `vmec_params::version_` = '8.52'
- integer `vmec_params::ntmax`
number of contributing Fourier basis function (can be 1, 2 or 4); assigned in `read_indata()`
- integer `vmec_params::rcc`
- integer `vmec_params::rss`
- integer `vmec_params::rsc`
- integer `vmec_params::rcs`
- integer `vmec_params::zsc`
- integer `vmec_params::zcs`
- integer `vmec_params::zcc`
- integer `vmec_params::zss`
- integer `vmec_params::mnyq`
- integer `vmec_params::nnyq`
- integer, dimension(:), allocatable `vmec_params::uminus`
- real(rprec), dimension(:), allocatable `vmec_params::mscale`
array for norming theta-trig functions (internal use only) so that the discrete $SUM[\cos(\mu)\cos(m'u)] = .5 \delta(m,m')$*
- real(rprec), dimension(:), allocatable `vmec_params::nscale`
array for norming zeta -trig functions (internal use only)
- real(rprec) `vmec_params::signgs`
sign of Jacobian : must be =1 (right-handed) or =-1 (left-handed)
- real(rprec) `vmec_params::lamscale` =1
- integer, parameter `vmec_params::m0` =0
from totzsp
- integer, parameter `vmec_params::m1` =1
from totzsp
- integer, parameter `vmec_params::n0` =0
from totzsp

7.27 src/data/vmec_persistent.f90 File Reference

Modules

- module `vmec_persistent`

Variables

- integer, dimension(:), allocatable `vmec_persistent::ixm`
- integer, dimension(:), allocatable `vmec_persistent::jmin3`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::cosmu`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::sinmu`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::cosmum`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::sinmum`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::cosmumi`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::sinmumi`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::cosnv`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::sinnv`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::cosnvn`
- real(rprec), dimension(:,:), allocatable `vmec_persistent::sinnvn`

- `real(rprec), dimension(:, :), allocatable vmec_persistent::cosmui`
- `real(rprec), dimension(:, :), allocatable vmec_persistent::sinmui`
- `real(rprec), dimension(:, :), allocatable vmec_persistent::cosmui3`
- `real(rprec), dimension(:, :), allocatable vmec_persistent::cosmumi3`
- `real(rprec), dimension(:), allocatable, target vmec_persistent::xm`
- `real(rprec), dimension(:), allocatable, target vmec_persistent::xn`
- `real(rprec), dimension(:), allocatable, target vmec_persistent::xm_nyq`
- `real(rprec), dimension(:), allocatable, target vmec_persistent::xn_nyq`
- `real(rprec), dimension(:), allocatable vmec_persistent::cos01`
- `real(rprec), dimension(:), allocatable vmec_persistent::sin01`

7.28 src/data/vmercier.f90 File Reference

Modules

- module `vmercier`

Variables

- `real(rprec), dimension(nsd) vmercier::dshear`
- `real(rprec), dimension(nsd) vmercier::dwell`
- `real(rprec), dimension(nsd) vmercier::dcurr`
- `real(rprec), dimension(nsd) vmercier::dmerc`
- `real(rprec), dimension(nsd) vmercier::dgeod`

7.29 src/data/vparams.f90 File Reference

Modules

- module `vparams`

Variables

- integer, parameter `vparams::nsd = 10001`
maximum number of radial nodes
- integer, parameter `vparams::mpold = 101`
maximum number of poloidal harmonics (in r,z,lam fourier series)
- integer, parameter `vparams::ntord = 101`
maximum number of toroidal harmonics
- integer, parameter `vparams::ndatafmax = 101`
- integer, parameter `vparams::nstore_seq = 100`
- integer, parameter `vparams::mpol1d = mpold - 1`
- integer, parameter `vparams::ntor1d = ntord + 1`
- integer, parameter `vparams::nthreed0 = 9`
- integer, parameter `vparams::indata0 = nthreed0 + 2`
- integer, parameter `vparams::nwout0 = nthreed0 + 3`
- integer, parameter `vparams::jxbout0 = nthreed0 + 4`
- integer, parameter `vparams::nfort18 = 18`

- integer, parameter `vparams::nmercier0` = 52
- integer `vparams::nthreed`
- real(rprec), parameter `vparams::c1pm2` = 1.e-2_dp
- real(rprec), parameter `vparams::cp15` = 0.15_dp
- real(rprec), parameter `vparams::cp25` = 0.25_dp
- real(rprec), parameter `vparams::cp5` = 0.50_dp
- real(rprec), parameter `vparams::c1pm8` = 1.0e-8_dp
- real(rprec), parameter `vparams::cbig` = 0.9e30_dp
- real(rprec), parameter `vparams::c2p0` = 2
- real(rprec), parameter `vparams::c3p0` = 3
- real(rprec), parameter `vparams::cp05` = 0.05_dp
- real(rprec), parameter `vparams::c1pm13` = 1.0e-13_dp
- real(rprec), parameter `vparams::osqrt2` = 0.707106781186547462_dp

7.30 src/data/vsvd0.f90 File Reference

Modules

- module `vsvd0`

Variables

- integer, parameter `vsvd0::nigroup` = 100
number of external current groups

7.31 src/data/xstuff.f90 File Reference

Modules

- module `xstuff`

Variables

- real(rprec), dimension(:), allocatable `xstuff::gc`
stacked array of R, Z, Lambda Spectral force coefficients (see above for stack order)
- real(rprec), dimension(:), allocatable, target `xstuff::xc`
stacked array of scaled R, Z, Lambda Fourier coefficients (see above for stack order)
- real(rprec), dimension(:), allocatable `xstuff::xcdot`
"velocity": change of Fourier coefficients per time step
- real(rprec), dimension(:), allocatable `xstuff::xsave`
- real(rprec), dimension(:), allocatable `xstuff::xstore`
backup copy of last-known-good xc
- real(rprec), dimension(:), allocatable `xstuff::scalxc`

7.32 src/elongation.f90 File Reference

Compute Waist thickness and height in $\varphi = 0, \pi$ symmetry planes.

Functions/Subroutines

- subroutine [elongation](#) (r1, z1, waist, height)
Compute Waist thickness and height in $\varphi = 0, \pi$ symmetry planes.

7.32.1 Detailed Description

Compute Waist thickness and height in $\varphi = 0, \pi$ symmetry planes.

7.32.2 Function/Subroutine Documentation

7.32.2.1 elongation()

```
subroutine elongation (
    real(rprec), dimension(ns,nzeta,ntheta3,0:1), intent(in) r1,
    real(rprec), dimension(ns,nzeta,ntheta3,0:1), intent(in) z1,
    real(rprec), dimension(2), intent(out) waist,
    real(rprec), dimension(2), intent(out) height )
```

Compute Waist thickness and height in $\varphi = 0, \pi$ symmetry planes.

Parameters

<i>r1</i>	<i>R</i>
<i>z1</i>	<i>Z</i>
<i>waist</i>	
<i>height</i>	

Definition at line 10 of file elongation.f90.

Referenced by [eqfor](#)().

Here is the caller graph for this function:



7.33 src/eqfor.f90 File Reference

Basis physics analysis and evaluation of force balance. This is where most of the contents of the `threed1` output file is computed.

Functions/Subroutines

- subroutine `eqfor` (`br`, `bz`, `bsubu`, `bsubv`, `tau`, `rzl_array`, `ier_flag`)

Basis physics analysis and evaluation of force balance. This is where most of the contents of the `threed1` output file is computed.

7.33.1 Detailed Description

Basis physics analysis and evaluation of force balance. This is where most of the contents of the `threed1` output file is computed.

7.33.2 Function/Subroutine Documentation

7.33.2.1 `eqfor()`

```
subroutine eqfor (
    real(rprec), dimension(nrzt), intent(out) br,
    real(rprec), dimension(nrzt), intent(out) bz,
    real(rprec), dimension(ns,nznt,0:1), intent(in) bsubu,
    real(rprec), dimension(ns,nznt,0:1), intent(in) bsubv,
    real(rprec), dimension(nrzt), intent(out) tau,
    real(rprec), dimension(ns,0:ntor,0:mpoll,3*ntmax), intent(in), target rzl_array,
    integer ier_flag )
```

Basis physics analysis and evaluation of force balance. This is where most of the contents of the `threed1` output file is computed.

Parameters

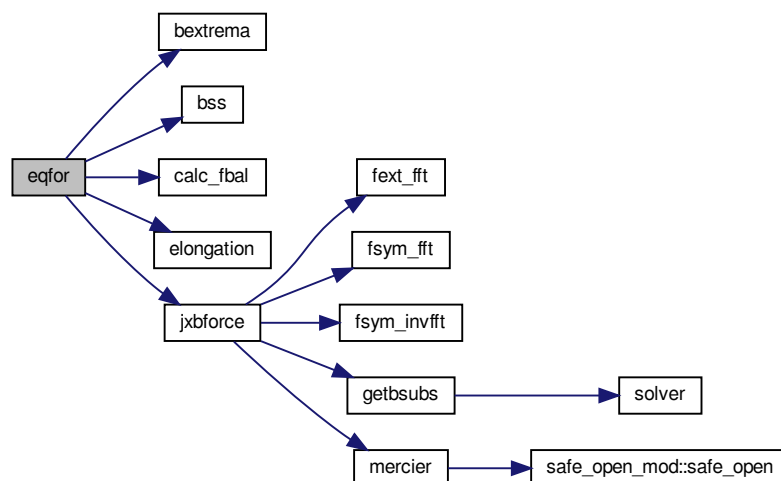
<code>br</code>	cylindrical component of magnetic field B^R
<code>bz</code>	cylindrical component of magnetic field B^Z
<code>bsubu</code>	covariant component of magnetic field B_θ
<code>bsubv</code>	covariant component of magnetic field B_ζ
<code>tau</code>	Jacobian $\sqrt{g} = R\tau$
<code>rzl_array</code>	state vector (all Fourier coefficients) of VMEC
<code>ier_flag</code>	error flag

Definition at line 15 of file `eqfor.f90`.

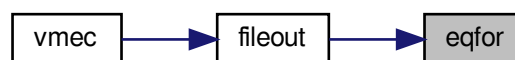
References vmec_io::aminor_p, vforces::armn, vforces::armn_o, vmec_main::aspect, vforces::azmn, vforces::azmn_e, vforces::azmn_o, vmec_io::b0, vmec_main::bdotb, vmec_main::beta_vol, vmec_io::betapol, vmec_io::betator, vmec_io::betatot, vmec_io::betaxis, bextrema(), vmec_main::bmax, vmec_main::bmin, vmec_main::bpedge, vacmod::bphiv, vmec_main::bledge, vforces::brmn, vforces::brmn_e, vacmod::bsqvac, bss(), vacmod::bsubvvac, vmec_main::buc, vmec_main::bucof, vmec_main::bvco, vmec_main::bvcof, vmec_main::bzedge, vforces::bzmn, vforces::bzmn_e, vforces::bzmn_o, calc_fbal(), vmec_main::chi, vmec_main::chipf, vmec_io::circum_p, vforces::crm_n_e, vforces::crm_n_o, vmec_io::cross_area_p, vmec_main::ctor, vforces::czmn_e, vforces::czmn_o, elongation(), vmec_main::equif, vmec_main::hs, vmec_io::ionlarmor, vmec_main::iotaf, vmec_main::iotas, vmec_main::ivac, vmec_main::jcuru, vmec_main::jcurv, vmec_main::jdotb, vmec_main::jpar2, vmec_main::jperp2, jxbforce(), vmec_io::kappa_p, vmec_main::lthreeed, vmec_params::mscale, vmec_params::nscale, vmec_params::ntmax, vmec_main::ovrr, realspace::phip, vmec_main::phipf, stel_constants::pi, vmec_main::pres, vmec_main::presf, vmec_main::presgrad, vmec_main::psi, vmec_main::r00, realspace::r1, vmec_main::rbtor, vmec_main::rbtor0, vmec_params::rcc, realspace::rcon, vmec_params::rcs, vmec_main::rinner, vmec_io::rmajor_p, vmec_io::rmax_surf, vmec_io::rmin_surf, vmec_main::router, realspace::ru0, realspace::rv, vmec_params::signgs, vmec_main::specw, realspace::sqrts, vmec_io::surf_area_p, vmec_io::volavgb, vmec_io::volume_p, vmec_main::vp, vmec_main::vpphi, realspace::wint, vmec_main::yellip, vmec_main::ygeo, vmec_main::yinden, vmec_main::yshift, vmec_main::ytrian, realspace::z1, vmec_params::zcc, realspace::zcon, vmec_params::zcs, vmec_io::zmax_surf, vmec_params::zsc, realspace::zu0, and realspace::zv.

Referenced by fileout().

Here is the call graph for this function:



Here is the caller graph for this function:



7.34 src/eqsolve.f90 File Reference

Iteratively evolve the Fourier coefficients that specify the equilibrium.

Functions/Subroutines

- subroutine `eqsolve` (`ier_flag`)
Iteratively evolve the Fourier coefficients that specify the equilibrium.

7.34.1 Detailed Description

Iteratively evolve the Fourier coefficients that specify the equilibrium.

7.34.2 Function/Subroutine Documentation

7.34.2.1 eqsolve()

```
subroutine eqsolve (
    integer, intent(inout) ier_flag )
```

Iteratively evolve the Fourier coefficients that specify the equilibrium.

Parameters

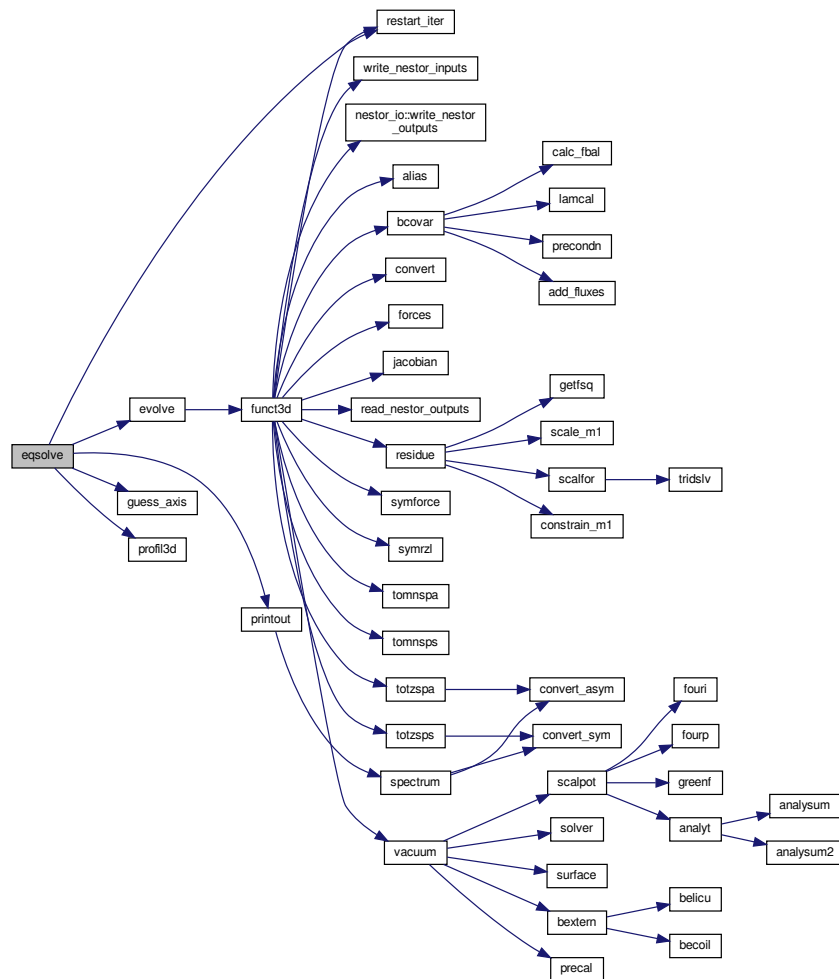
<code>ier_flag</code>	error flag
-----------------------	------------

Definition at line 7 of file eqsolve.f90.

References `vmec_params::bad_jacobian_flag`, `vmec_main::delt0r`, `evolve()`, `vmec_main::fsq`, `vmec_main::fsqr`, `vmec_main::fsqz`, `guess_axis()`, `vmec_main::ijacob`, `vmec_main::irst`, `vmec_main::irzloff`, `vmec_main::iter1`, `vmec_main::iter2`, `vmec_main::ivac`, `vmec_params::jac75_flag`, `vmec_main::niterv`, `vmec_params::norm_term_flag`, `vmec_params::ns4`, `printout()`, `profil3d()`, `realspace::r1`, `vmec_main::res0`, `restart_iter()`, `realspace::ru0`, `vmec_params::successful_term_flag`, `vmec_main::wb`, `vmec_main::wp`, `xstuff::xc`, `realspace::z1`, and `realspace::zu0`.

Referenced by `vmec()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.35 src/evolve.f90 File Reference

Take a single time step in Fourier space to evolve the Fourier coefficients describing the equilibrium towards force balance.

Functions/Subroutines

- subroutine `evolve` (`time_step`, `ier_flag`, `liter_flag`)

Take a single time step in Fourier space to evolve the Fourier coefficients describing the equilibrium towards force balance.

7.35.1 Detailed Description

Take a single time step in Fourier space to evolve the Fourier coefficients describing the equilibrium towards force balance.

7.35.2 Function/Subroutine Documentation

7.35.2.1 `evolve()`

```
subroutine evolve (
    real(rprec), intent(in) time_step,
    integer, intent(inout) ier_flag,
    logical, intent(inout) liter_flag )
```

Take a single time step in Fourier space to evolve the Fourier coefficients describing the equilibrium towards force balance.

Parameters

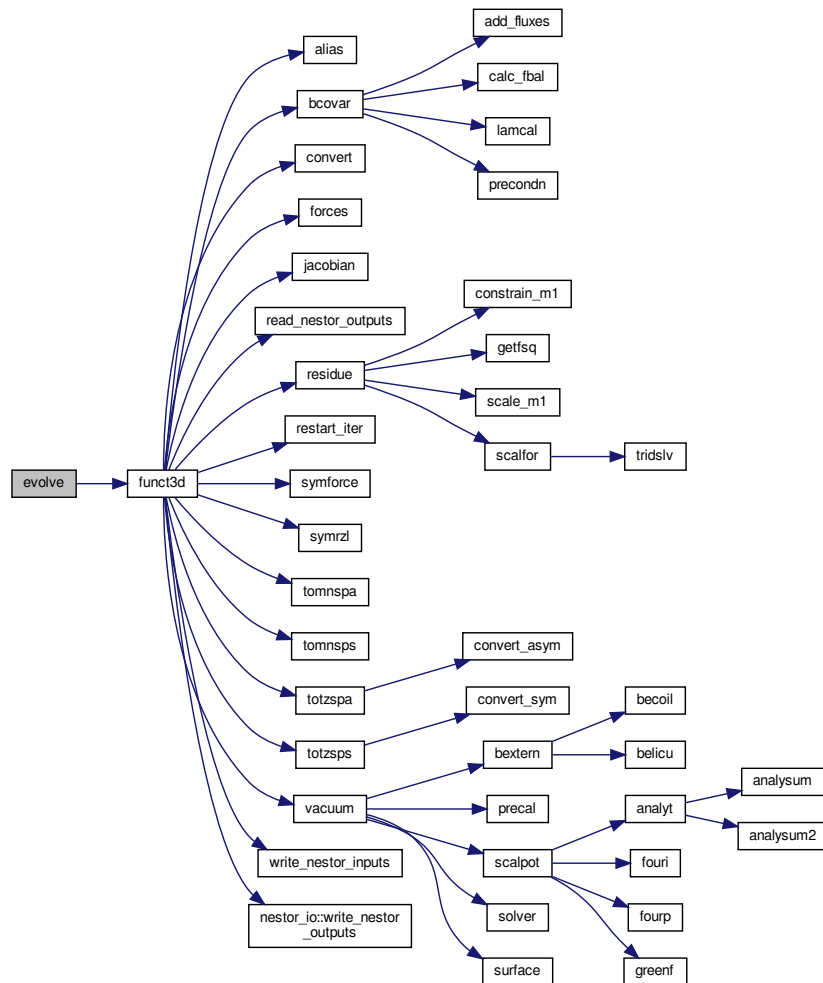
<i>time_step</i>	step length in parameter space to take
<i>ier_flag</i>	error flag
<i>liter_flag</i>	keep running?

Definition at line 11 of file evolve.f90.

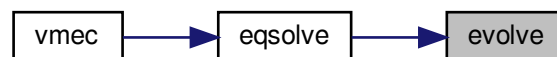
References `vmec_params::bad_jacobian_flag`, `vmec_main::fsq`, `vmec_main::fsql`, `vmec_main::fsql1`, `vmec_main::fsqr`, `vmec_main::fsqr1`, `vmec_main::fsqz`, `vmec_main::fsqz1`, `vmec_main::ftolv`, `funct3d()`, `xstuff::gc`, `vmec_main::irst`, `vmec_main::iter1`, `vmec_main::iter2`, `vmec_params::norm_term_flag`, `vmec_main::otau`, `vmec_main::otav`, `vmec_params::successful_term_flag`, `xstuff::xc`, and `xstuff::xcdot`.

Referenced by `eqsolve()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.36 src/fileout.f90 File Reference

Write the output files.

Functions/Subroutines

- subroutine `fileout` (`ier_flag`)
Write the output files.

7.36.1 Detailed Description

Write the output files.

7.36.2 Function/Subroutine Documentation

7.36.2.1 `fileout()`

```
subroutine fileout (
    integer, intent(inout) ier_flag )
```

Write the output files.

Parameters

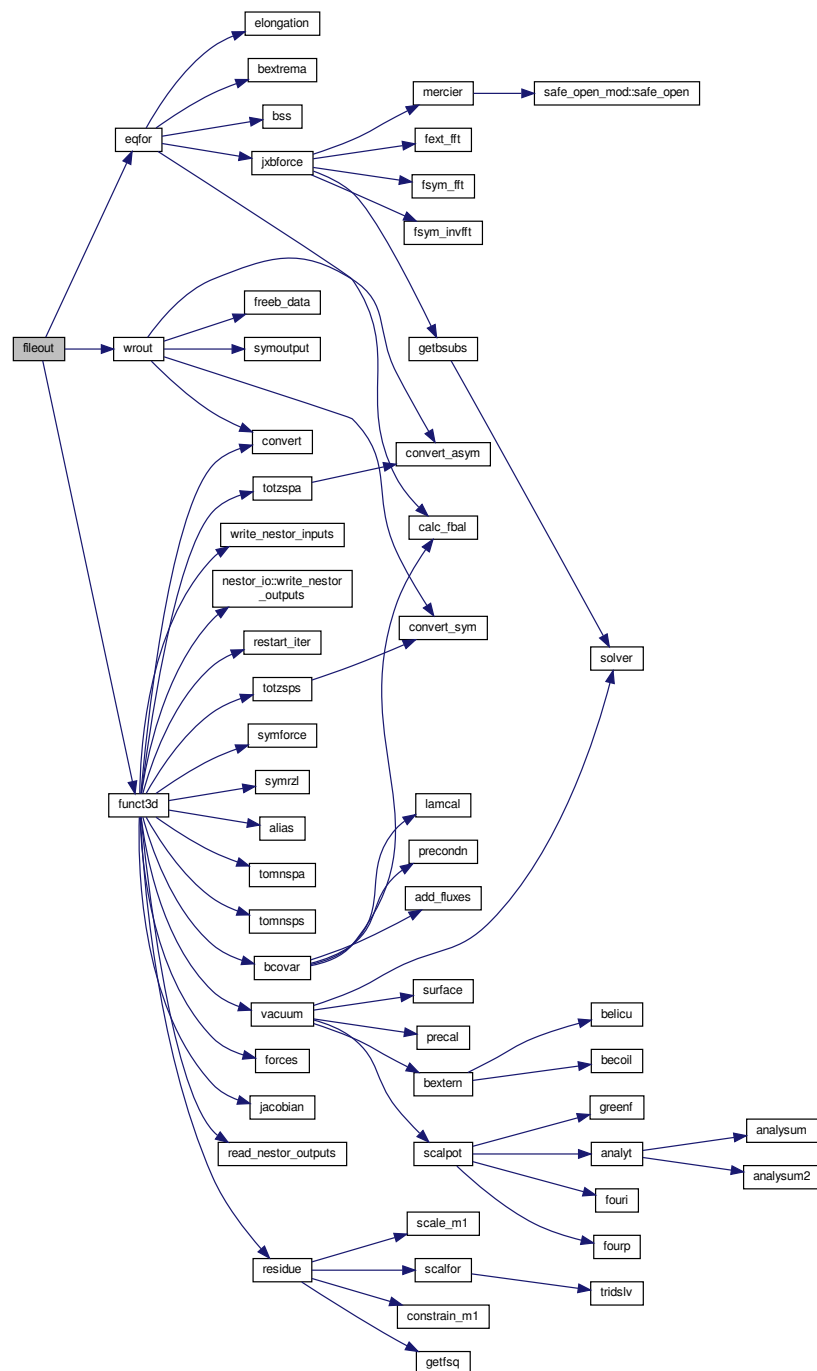
<code>ier_flag</code>	error flag
-----------------------	------------

Definition at line 7 of file fileout.f90.

References `vforces::azmn_o`, `vforces::blmn`, `vforces::bzmno`, `vforces::clmn`, `vforces::crmn_e`, `vforces::crmn_o`, `vforces::czmn_e`, `eqfor()`, `funct3d()`, `xstuff::gc`, `vmec_main::hs`, `vmec_main::iequi`, `vmec_main::ijacob`, `vmec_main::irst`, `vmec_params::norm_term_flag`, `vmec_main::phi`, `realspace::phip`, `realspace::rcon`, `vmec_params::signs`, `vmec_params::successful_term_flag`, `wrout()`, `xstuff::xc`, and `xstuff::xsave`.

Referenced by `vmec()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.37 src/fixaray.f90 File Reference

allocate and fill some fixed-size arrays (only depending on Fourier resolution).

Functions/Subroutines

- subroutine [fixaray](#)
allocate and fill some fixed-size arrays (only depending on Fourier resolution).

7.37.1 Detailed Description

allocate and fill some fixed-size arrays (only depending on Fourier resolution).

7.37.2 Function/Subroutine Documentation

7.37.2.1 fixaray()

```
subroutine fixaray
```

allocate and fill some fixed-size arrays (only depending on Fourier resolution).

Definition at line 6 of file fixaray.f90.

References `vmec_main::faccon`, `vmec_params::jmin2`, `vmec_params::mnyq`, `vmec_params::mscale`, `vmec_↔
params::nnyq`, `vmec_params::nscale`, `vmec_main::r0scale`, `vmec_params::signgs`, and `vmec_main::xmpq`.

Referenced by `vmec()`.

Here is the caller graph for this function:



7.38 src/flip_theta.f90 File Reference

Flip the definition of the poloidal angle in the user-provided initial guess for the LCFS geometry.

Functions/Subroutines

- subroutine [flip_theta](#) (rmn, zmn, lmn)

Flip the definition of the poloidal angle in the user-provided initial guess for the LCFS geometry.

7.38.1 Detailed Description

Flip the definition of the poloidal angle in the user-provided initial guess for the LCFS geometry.

7.38.2 Function/Subroutine Documentation

7.38.2.1 flip_theta()

```
subroutine flip_theta (
    real(rprec), dimension(0:ntor,0:mpoll,ntmax), intent(inout) rmn,
    real(rprec), dimension(0:ntor,0:mpoll,ntmax), intent(inout) zmn,
    real(rprec), dimension(0:ntor,0:mpoll,ntmax), intent(inout), optional lmn )
```

Flip the definition of the poloidal angle in the user-provided initial guess for the LCFS geometry.

Parameters

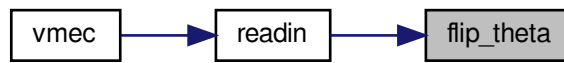
	<i>rmn</i>	Fourier coefficients for R
	<i>zmn</i>	Fourier coefficients for Z
	<i>lmn</i>	Fourier coefficients for λ
<i>in, out</i>	<i>lmn</i>	never used: can also flip lambda...

Definition at line 9 of file flip_theta.f90.

References vmec_main::lthreed, vmec_params::ntmax, vmec_params::rcc, vmec_params::rcs, vmec_params::rsc, vmec_params::rss, vmec_params::zcc, vmec_params::zcs, vmec_params::zsc, and vmec_params::zss.

Referenced by readin().

Here is the caller graph for this function:



7.39 src/forces.f90 File Reference

Compute the real-space MHD forces.

Functions/Subroutines

- subroutine [forces](#)
Compute the real-space MHD forces.

7.39.1 Detailed Description

Compute the real-space MHD forces.

7.39.2 Function/Subroutine Documentation

7.39.2.1 forces()

```
subroutine forces
```

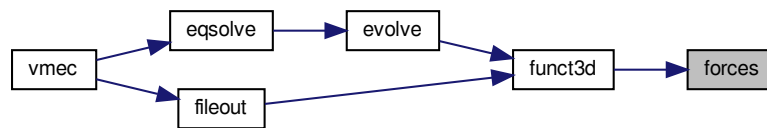
Compute the real-space MHD forces.

Definition at line 6 of file forces.f90.

References `vforces::armn_e`, `vforces::armn_o`, `vforces::azmn_e`, `vforces::azmn_o`, `vforces::brmn_e`, `vforces::brmn_o`, `vforces::bzmne`, `vforces::bzmno`, `vforces::crmn_e`, `vforces::crmn_o`, `vforces::czmn_e`, `vforces::czmn_o`, `realspace::extra1`, `realspace::extra2`, `realspace::extra3`, `realspace::extra4`, `realspace::gcon`, `realspace::guu`, `realspace::guv`, `realspace::gvv`, `vmec_main::ivac`, `vmec_main::lthreed`, `vmec_main::ohs`, `realspace::r1`, `vmec_main::rbsq`, `realspace::rcon`, `realspace::rcon0`, `realspace::ru`, `realspace::ru0`, `realspace::rv`, `realspace::shalf`, `realspace::sqrts`, `realspace::z1`, `realspace::zcon`, `realspace::zcon0`, `realspace::zu`, `realspace::zu0`, and `realspace::zv`.

Referenced by `funct3d()`.

Here is the caller graph for this function:



7.40 src/free_mem_funct3d.f90 File Reference

Free memory required by [funct3d\(\)](#)

Functions/Subroutines

- subroutine [free_mem_funct3d](#)
Free memory required by [funct3d\(\)](#)

7.40.1 Detailed Description

Free memory required by [funct3d\(\)](#)

7.40.2 Function/Subroutine Documentation

7.40.2.1 free_mem_funct3d()

```
subroutine free_mem_funct3d
```

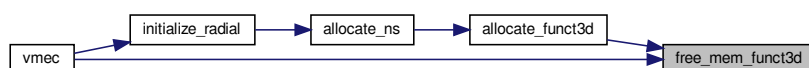
Free memory required by [funct3d\(\)](#)

Definition at line 6 of file free_mem_funct3d.f90.

References `vforces::armn`, `vforces::azmn`, `vforces::blmn`, `vforces::brmn`, `vforces::bzmn`, `vforces::clmn`, `vforces::crmn`, `vforces::czmn`, `realspace::extra1`, `realspace::extra2`, `realspace::extra3`, `realspace::extra4`, `realspace::gcon`, `realspace::guu`, `realspace::guv`, `realspace::gvv`, `realspace::r1`, `realspace::rcon`, `realspace::rcon0`, `realspace::ru`, `realspace::ru0`, `realspace::rv`, `realspace::z1`, `realspace::zcon`, `realspace::zcon0`, `realspace::zu`, `realspace::zu0`, and `realspace::zv`.

Referenced by `allocate_funct3d()`, and `vmec()`.

Here is the caller graph for this function:



7.41 src/free_mem_ns.f90 File Reference

Free memory depending on the number of flux surfaces `ns`.

Functions/Subroutines

- subroutine [free_mem_ns](#)

Free memory depending on the number of flux surfaces `ns`.

7.41.1 Detailed Description

Free memory depending on the number of flux surfaces `ns`.

7.41.2 Function/Subroutine Documentation

7.41.2.1 free_mem_ns()

```
subroutine free_mem_ns
```

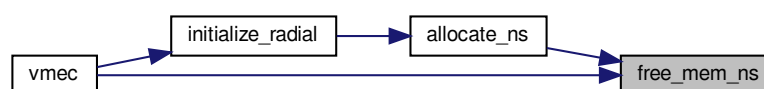
Free memory depending on the number of flux surfaces `ns`.

Definition at line 6 of file `free_mem_ns.f90`.

References `vmec_main::ard`, `vmec_main::arm`, `vmec_main::azd`, `vmec_main::azm`, `vmec_main::bdamp`, `vmec_main::bdotb`, `vmec_main::bdotgradv`, `vmec_main::beta_vol`, `vmec_main::blam`, `vmec_main::bmax`, `vmec_main::bmin`, `vmec_main::brd`, `vmec_main::brm`, `vmec_main::buco`, `vmec_main::bucof`, `vmec_main::bvco`, `vmec_main::bvcof`, `vmec_main::bzd`, `vmec_main::bzm`, `vmec_main::chi`, `realspace::chip`, `vmec_main::chipf`, `vmec_main::chips`, `vmec_main::clam`, `vmec_main::crd`, `vmec_main::dlam`, `vmec_main::equiv`, `vmec_main::faclam`, `fbal::frcc_fac`, `fbal::fzsc_fac`, `xstuff::gc`, `vmec_main::icurv`, `vmec_main::iotaf`, `vmec_main::iotas`, `vmec_main::ireflect`, `vmec_main::jcuru`, `vmec_main::jcurv`, `vmec_main::jdotb`, `vmec_main::jpar2`, `vmec_main::jperp2`, `vmec_main::mass`, `vmec_main::overr`, `vmec_main::phi`, `realspace::phip`, `vmec_main::phipf`, `vmec_main::phips`, `vmec_main::pres`, `vmec_main::presf`, `vmec_main::presgrad`, `vmec_main::psi`, `fbal::rru_fac`, `fbal::rzu_fac`, `xstuff::scalxc`, `realspace::shalf`, `vmec_main::sm`, `vmec_main::sp`, `vmec_main::specw`, `realspace::sqrts`, `vmec_main::tcon`, `vmec_main::vp`, `vmec_main::vpphi`, `realspace::wint`, `xstuff::xc`, `xstuff::xcdot`, `xstuff::xsave`, `xstuff::xstore`, `vmec_main::yellip`, `vmec_main::ygeo`, `vmec_main::yinden`, `vmec_main::yshift`, and `vmec_main::ytrian`.

Referenced by `allocate_ns()`, and `vmec()`.

Here is the caller graph for this function:



7.42 src/free_mem_nunv.f90 File Reference

Free arrays depending on the number of Fourier coefficients `nunv`.

Functions/Subroutines

- subroutine `free_mem_nunv`

Free arrays depending on the number of Fourier coefficients `nunv`.

7.42.1 Detailed Description

Free arrays depending on the number of Fourier coefficients `nunv`.

7.42.2 Function/Subroutine Documentation

7.42.2.1 `free_mem_nunv()`

```
subroutine free_mem_nunv
```

Free arrays depending on the number of Fourier coefficients `nunv`.

Definition at line 6 of file `free_mem_nunv.f90`.

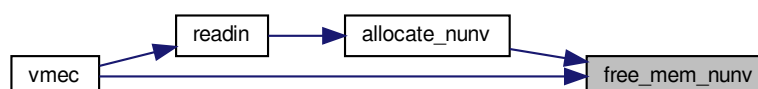
References `vmec_main::bsqsav`, `vmec_main::bsubu0`, `vmec_main::dbsq`, `vacmod::free_mem_nestor()`, `vmec_main::rbsq`, `vmec_main::rmn_bdy`, and `vmec_main::zmn_bdy`.

Referenced by `allocate_nunv()`, and `vmec()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.43 src/freeb_data.f90 File Reference

Write out edge values of fields.

Functions/Subroutines

- subroutine [freeb_data](#) (rmnc, zmns, rmns, zmnc, bmodmn, bmodmn1)

Write out edge values of fields.

7.43.1 Detailed Description

Write out edge values of fields.

7.43.2 Function/Subroutine Documentation

7.43.2.1 freeb_data()

```
subroutine freeb_data (
    real(rprec), dimension(mnmax) rmnc,
    real(rprec), dimension(mnmax) zmns,
    real(rprec), dimension(mnmax) rmns,
    real(rprec), dimension(mnmax) zmnc,
    real(rprec), dimension(mnmax) bmodmn,
    real(rprec), dimension(mnmax) bmodmn1 )
```

Write out edge values of fields.

Parameters

<i>rmnc</i>	stellarator-symmetric Fourier coefficients of R
<i>zmns</i>	stellarator-symmetric Fourier coefficients of Z
<i>rmns</i>	non-stellarator-symmetric Fourier coefficients of R
<i>zmnc</i>	non-stellarator-symmetric Fourier coefficients of Z
<i>bmodmn</i>	stellarator-symmetric Fourier coefficients of $ \mathbf{B} $
<i>bmodmn1</i>	non-stellarator-symmetric Fourier coefficients of $ \mathbf{B} $

Definition at line 12 of file freeb_data.f90.

References vmec_main::bpedge, vacmod::bphiv, vmec_main::bledge, vacmod::brv, vmec_main::bsqsav, vacmod::bsqvac, vmec_main::bzedge, vacmod::bzv, vmec_main::ivac, vacmod::potvac, realspace::r1, and realspace::z1.

Referenced by wrout().

Here is the caller graph for this function:



7.44 src/fsym_fft.f90 File Reference

Fourier transforms.

Functions/Subroutines

- subroutine `fext_fft` (bout, bs_s, bs_a)
Extends B_s from $n\theta_{a2}$ interval to full $n\theta_{a3}$ interval in angle θ .
- subroutine `fsym_fft` (bs, bu, bv, bs_s, bu_s, bv_s, bs_a, bu_a, bv_a)
Contract bs,bu,bv from full nu interval to half- u interval so cos, sin integrals can be performed on half- u interval.

7.44.1 Detailed Description

Fourier transforms.

7.44.2 Function/Subroutine Documentation

7.44.2.1 `fext_fft()`

```

subroutine fext_fft (
    real(rprec), dimension(nzeta,ntheta3), intent(out) bout,
    real(rprec), dimension(nzeta,ntheta2), intent(in) bs_s,
    real(rprec), dimension(nzeta,ntheta2), intent(in) bs_a )

```

Extends B_s from $n\theta_{a2}$ interval to full $n\theta_{a3}$ interval in angle θ .

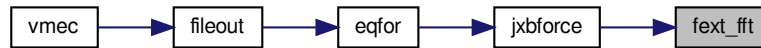
Parameters

<i>bout</i>	output B_s
<i>bs</i> _↔ <i>_s</i>	symmetric part of B_s
<i>bs</i> _↔ <i>_a</i>	anti-symmetric part of B_s

Definition at line 9 of file fsym_fft.f90.

Referenced by jxbforce().

Here is the caller graph for this function:



7.44.2.2 fsym_fft()

```

subroutine fsym_fft (
    real(rprec), dimension(nzeta,ntheta3), intent(in) bs,
    real(rprec), dimension(nzeta,ntheta3,0:1), intent(in) bu,
    real(rprec), dimension(nzeta,ntheta3,0:1), intent(in) bv,
    real(rprec), dimension(nzeta,ntheta2) bs_s,
    real(rprec), dimension(nzeta,ntheta2,0:1), intent(out) bu_s,
    real(rprec), dimension(nzeta,ntheta2,0:1), intent(out) bv_s,
    real(rprec), dimension(nzeta,ntheta2) bs_a,
    real(rprec), dimension(nzeta,ntheta2,0:1), intent(out) bu_a,
    real(rprec), dimension(nzeta,ntheta2,0:1), intent(out) bv_a )

```

Contract bs,bu,bv from full nu interval to half-u interval so cos, sin integrals can be performed on half-u interval.

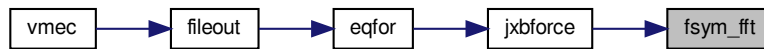
Parameters

<i>bs</i>	output B_s
<i>bu</i>	output B_θ
<i>bv</i>	output B_{zeta}
<i>bs</i> _↔ <i>_s</i>	symmetric part of B_s
<i>bu</i> _↔ <i>_s</i>	symmetric part of B_θ
<i>bv</i> _↔ <i>_s</i>	symmetric part of B_ζ
<i>bs</i> _↔ <i>_a</i>	anti-symmetric part of B_s
<i>bu</i> _↔ <i>_a</i>	anti-symmetric part of B_θ
<i>bv</i> _↔ <i>_a</i>	anti-symmetric part of B_ζ

Definition at line 47 of file fsym_fft.f90.

Referenced by jxbforce().

Here is the caller graph for this function:



7.45 src/fsym_invfft.f90 File Reference

Extends function from `ntheta2` to `ntheta3` range.

Functions/Subroutines

- subroutine `fsym_invfft` (`bsubsu`, `bsubsv`)
Extends function from `ntheta2` to `ntheta3` range.

7.45.1 Detailed Description

Extends function from `ntheta2` to `ntheta3` range.

7.45.2 Function/Subroutine Documentation

7.45.2.1 `fsym_invfft()`

```

subroutine fsym_invfft (
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) bsubsu,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) bsubsv )

```

Extends function from `ntheta2` to `ntheta3` range.

Parameters

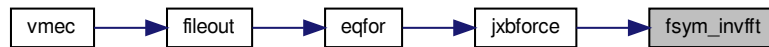
<i>bsubsu</i>	tangential derivative of covariant magnetic field component $\partial B_s / \partial \theta$
<i>bsubsv</i>	tangential derivative of covariant magnetic field component $\partial B_s / \partial \zeta$

Definition at line 8 of file `fsym_invfft.f90`.

References `vmec_main::ireflect`.

Referenced by `jxbforce()`.

Here is the caller graph for this function:



7.46 src/funct3d.f90 File Reference

Evaluate the three-dimensional MHD energy functional.

Functions/Subroutines

- subroutine `funct3d` (`ier_flag`)
Evaluate the three-dimensional MHD energy functional.

7.46.1 Detailed Description

Evaluate the three-dimensional MHD energy functional.

7.46.2 Function/Subroutine Documentation

7.46.2.1 funct3d()

```

subroutine funct3d (
    integer, intent(inout) ier_flag )

```

Evaluate the three-dimensional MHD energy functional.

Parameters

<code>ier_flag</code>	error flag
-----------------------	------------

use system call to stand-alone NESTOR for vacuum computation

dump reference input for and output of NESTOR when using internal NESTOR

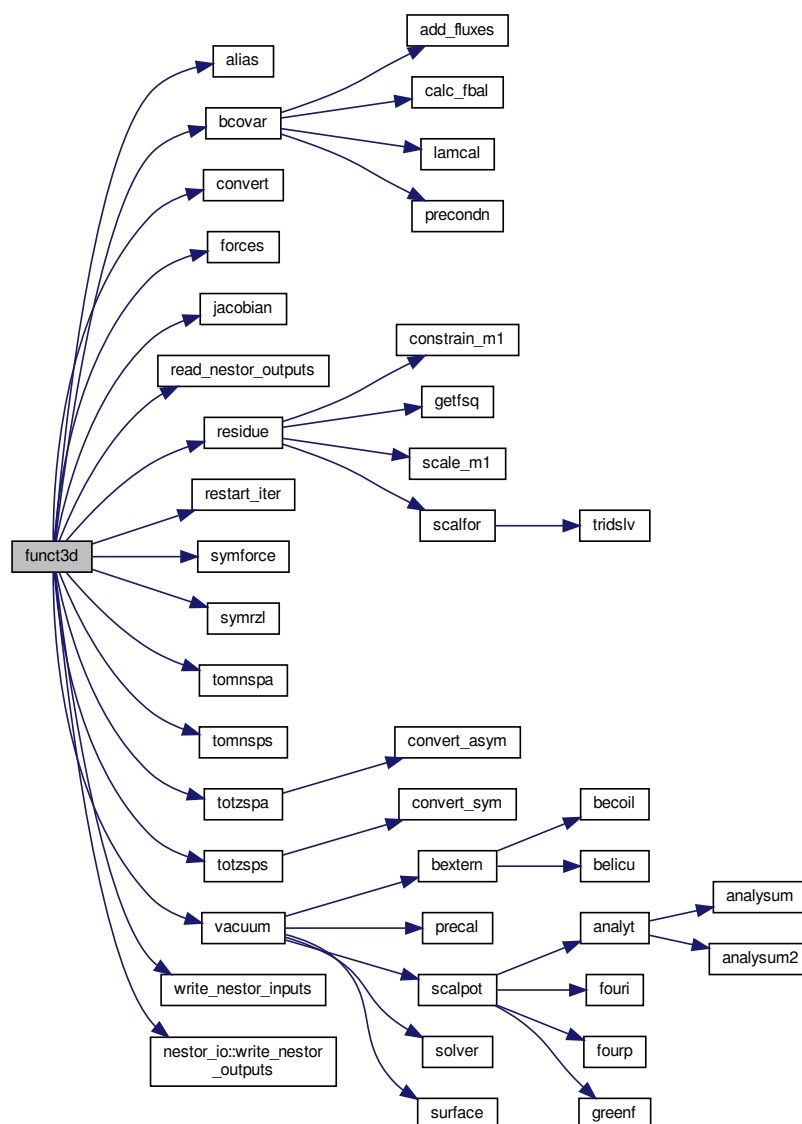
Definition at line 7 of file funct3d.f90.

References `alias()`, `vacmod::amatsav`, `vforces::armn`, `vforces::azmn`, `vmec_params::bad_jacobian_flag`, `bco-var()`, `vforces::blmn`, `vforces::brmn`, `vmec_main::bsqsav`, `vacmod::bsqvac`, `vacmod::bsubvvac`, `vacmod::bvecsav`,

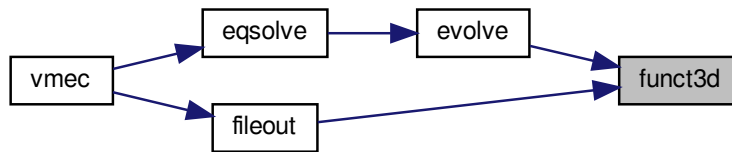
vforces::bzm, vforces::bzm_o, vforces::clm, convert(), vforces::crm, vmec_main::ctor, vforces::czm, vmec_main::dbsq, realspace::extra1, realspace::extra2, realspace::extra3, realspace::extra4, forces(), vmec_main::fsq, vmec_main::fsqr, vmec_main::fsqz, xstuff::gc, realspace::gcon, vmec_main::hs, vmec_main::iequi, vmec_main::irst, vmec_main::irzloff, vmec_main::iter1, vmec_main::iter2, vmec_main::ivac, jacobian(), vmec_main::neqs, vmec_main::ohs, vmec_main::pres, vmec_main::r00, realspace::r1, vmec_main::rbsq, vmec_main::rbtor, realspace::rcon, realspace::rcon0, read_nestor_outputs(), residue(), restart_iter(), vmec_main::rinner, vmec_main::router, realspace::ru, realspace::ru0, realspace::rv, xstuff::scalxc, vmec_params::signgs, realspace::sqrts, symforce(), symrzl(), tomnspace(), tomnsps(), totzsps(), totzsps(), vacuum(), vmec_main::vacuum_calls, realspace::wint, write_nestor_inputs(), nestor_io::write_nestor_outputs(), xstuff::xc, vmec_main::z00, realspace::z1, realspace::zcon, realspace::zcon0, realspace::zu, realspace::zu0, and realspace::zv.

Referenced by evolve(), and fileout().

Here is the call graph for this function:



Here is the caller graph for this function:



7.47 src/functions.f File Reference

This module contains functions used by the profiles.

Modules

- module [functions](#)

Functions/Subroutines

- real(rprec) function, public [functions::two_power](#) (x, b)
*Profile function for the two_power profile. $b(0) * (1 - x^{b(1)})^{b(2)}$.*
- real(rprec) function, public [functions::two_power_gs](#) (x, b)
*Profile function for the two_power_gs profile. $two_power(x) * (1 + \sum [b(i) * \exp(-(x - b(i + 1))/b(i + 2))^2])$.*
- logical function [functions::function_test](#) ()
Main test function.

7.47.1 Detailed Description

This module contains functions used by the profiles.

7.48 src/getbsubs.f90 File Reference

Solves the radial force balance $\mathbf{B} \cdot \mathbf{B}_s = F_s$ for B_s in real space using collocation.

Functions/Subroutines

- subroutine [getbsubs](#) (bsubsmn, frho, bsupu, bsupv, mmax, nmax, info)
Solves the radial force balance $\mathbf{B} \cdot \mathbf{B}_s = F_s$ for B_s in real space using collocation.

7.48.1 Detailed Description

Solves the radial force balance $\mathbf{B} \cdot B_s = F_s$ for B_s in real space using collocation.

7.48.2 Function/Subroutine Documentation

7.48.2.1 getbsubs()

```
subroutine getbsubs (
    real(rprec), dimension(0:mmax, -nmax:nmax, 0:1), intent(out) bsubsmn,
    real(rprec), dimension(nzeta, ntheta3), intent(in) frho,
    real(rprec), dimension(nzeta, ntheta3), intent(in) bsupu,
    real(rprec), dimension(nzeta, ntheta3), intent(in) bsupv,
    integer, intent(in) mmax,
    integer, intent(in) nmax,
    integer, intent(out) info )
```

Solves the radial force balance $\mathbf{B} \cdot B_s = F_s$ for B_s in real space using collocation.

Parameters

<i>bsubsmn</i>	Fourier coefficients of B_s
<i>frho</i>	Fourier coefficients of radial Force component
<i>bsupu</i>	contravariant component of magnetic field B^θ
<i>bsupv</i>	contravariant component of magnetic field B^ζ
<i>mmax</i>	maximum poloidal mode number
<i>nmax</i>	maximum toroidal mode number
<i>info</i>	error flag

Definition at line 13 of file getbsubs.f90.

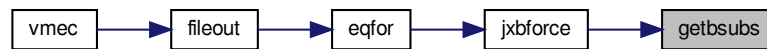
References vmec_persistent::cosmu, vmec_persistent::cosnv, vmec_input::lasym, vmec_input::nfp, vmec_dim↔
::ntheta2, vmec_dim::ntheta3, vmec_input::nzeta, vmec_persistent::sinmu, vmec_persistent::sinnv, and solver().

Referenced by jxbforce().

Here is the call graph for this function:



Here is the caller graph for this function:



7.49 src/getcurmid.f90 File Reference

Get current at midplane (?)

Functions/Subroutines

- subroutine [getcurmid](#) (curmid, izeta, gsqrt, r12)
Get current at midplane (?)

7.49.1 Detailed Description

Get current at midplane (?)

7.49.2 Function/Subroutine Documentation

7.49.2.1 getcurmid()

```

subroutine getcurmid (
    real(rprec), dimension(2*ns) curmid,
    real(rprec), dimension(ns,nzeta,*) izeta,
    real(rprec), dimension(ns,nzeta,*) gsqrt,
    real(rprec), dimension(ns,nzeta,*) r12 )

```

Get current at midplane (?)

Parameters

<i>curmid</i>	current at midplane (?)
<i>izeta</i>	index in toroidal direction
<i>gsqrt</i>	Jacobian
<i>r12</i>	R^2

Definition at line 10 of file getcurmid.f90.

References vmec_dim::ns, vmec_dim::ns1, vmec_dim::ntheta2, and vmec_input::nzeta.

7.50 src/getfsq.f90 File Reference

Compute total force residual on flux surfaces.

Functions/Subroutines

- subroutine [getfsq](#) (gcr, gcz, gnormr, gnormz, gnorm, medge)
Compute total force residual on flux surfaces.

7.50.1 Detailed Description

Compute total force residual on flux surfaces.

7.50.2 Function/Subroutine Documentation

7.50.2.1 getfsq()

```
subroutine getfsq (
    real(rprec), dimension(ns,mnsize*ntmax), intent(in) gcr,
    real(rprec), dimension(ns,mnsize*ntmax), intent(in) gcz,
    real(rprec), intent(out) gnormr,
    real(rprec), intent(out) gnormz,
    real(rprec), intent(in) gnorm,
    integer, intent(in) medge )
```

Compute total force residual on flux surfaces.

Parameters

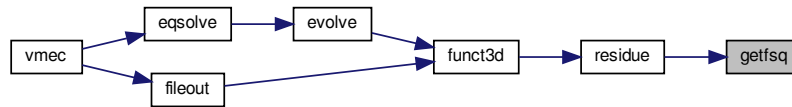
<i>gcr</i>	<i>R</i> -component of force
<i>gcz</i>	<i>Z</i> -component of force
<i>gnormr</i>	normalized total force residual in <i>R</i>
<i>gnormz</i>	normalized total force residual in <i>Z</i>
<i>gnorm</i>	normalization factor for forces
<i>medge</i>	=0: exclude contribution from LCFS; =1: include LCFS contribution

Definition at line 12 of file getfsq.f90.

References vmec_params::ntmax.

Referenced by residue().

Here is the caller graph for this function:



7.51 src/guess_axis.f90 File Reference

Computes guess for magnetic axis if user guess leads to initial sign change of Jacobian.

Functions/Subroutines

- subroutine [guess_axis](#) (*r1*, *z1*, *ru0*, *zu0*)

Computes guess for magnetic axis if user guess leads to initial sign change of Jacobian.

7.51.1 Detailed Description

Computes guess for magnetic axis if user guess leads to initial sign change of Jacobian.

7.51.2 Function/Subroutine Documentation

7.51.2.1 guess_axis()

```

subroutine guess_axis (
    real(rprec), dimension(ns,nzeta,ntheta3,0:1), intent(in) r1,
    real(rprec), dimension(ns,nzeta,ntheta3,0:1), intent(in) z1,
    real(rprec), dimension(ns,nzeta,ntheta3), intent(in) ru0,
    real(rprec), dimension(ns,nzeta,ntheta3), intent(in) zu0 )

```

Computes guess for magnetic axis if user guess leads to initial sign change of Jacobian.

Parameters

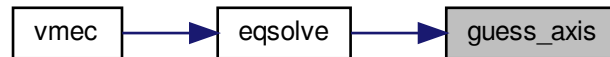
<i>r1</i>	<i>R</i>
<i>z1</i>	<i>Z</i>
<i>ru0</i>	$\partial R / \partial \theta$
<i>zu0</i>	$\partial Z / \partial \theta$

Definition at line 10 of file guess_axis.f90.

References vmec_main::hs, vmec_params::nscale, vmec_params::signgs, and realspace::sqrts.

Referenced by eqsolve().

Here is the caller graph for this function:



7.52 src/heading.f90 File Reference

Open output files and print banner message at the top.

Functions/Subroutines

- subroutine [heading](#) (extension)
Open output files and print banner message at the top.

7.52.1 Detailed Description

Open output files and print banner message at the top.

7.52.2 Function/Subroutine Documentation

7.52.2.1 heading()

```

subroutine heading (
    character(len=*), intent(in) extension )
  
```

Open output files and print banner message at the top.

Parameters

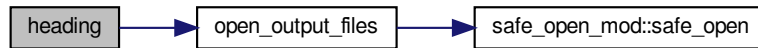
<i>extension</i>	input file "extension": part after 'input.'.
------------------	--

Definition at line 7 of file heading.f90.

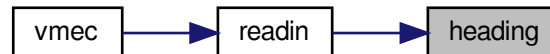
References vparams::nthreed, open_output_files(), and vmec_params::version_.

Referenced by readin().

Here is the call graph for this function:



Here is the caller graph for this function:



7.53 src/initialize_radial.f90 File Reference

Allocates memory for radial arrays and initializes radial profiles.

Functions/Subroutines

- subroutine [initialize_radial](#) (nsval, ns_old, delt0)
Allocates memory for radial arrays and initializes radial profiles.

7.53.1 Detailed Description

Allocates memory for radial arrays and initializes radial profiles.

7.53.2 Function/Subroutine Documentation

7.53.2.1 initialize_radial()

```

subroutine initialize_radial (
    integer, intent(in) nsval,
    integer, intent(inout) ns_old,
    real(rprec), intent(out) delt0 )
  
```

Allocates memory for radial arrays and initializes radial profiles.

Parameters

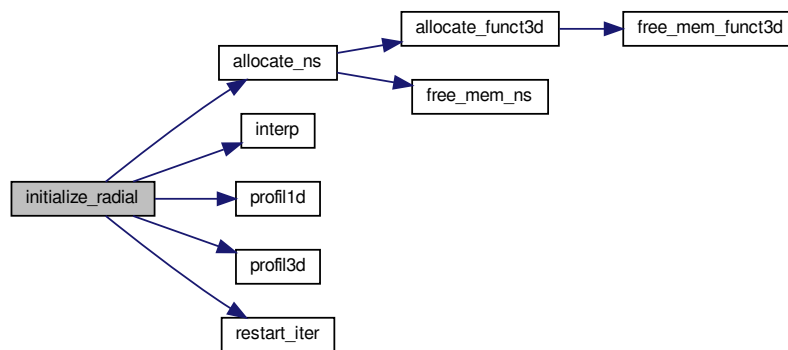
<i>nsval</i>	new number of flux surfaces
<i>ns_old</i>	old number of flux surfaces (from previous multi-grid iteration)
<i>delt0</i>	time step to be used in the new multi-grid iteration

Definition at line 9 of file initialize_radial.f90.

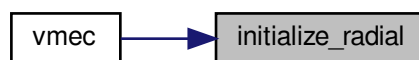
References `allocate_ns()`, `vmec_main::fsq`, `vmec_main::ftolv`, `xstuff::gc`, `vmec_main::hs`, `vmec_main::ijacob`, `interp()`, `vmec_main::irst`, `vmec_main::irzloff`, `vmec_main::iter1`, `vmec_main::iter2`, `vmec_main::neqs`, `vmec_main::niterv`, `vmec_params::ntmax`, `vmec_main::ohs`, `profil1d()`, `profil3d()`, `vmec_main::res0`, `restart_iter()`, `xstuff::scalxc`, `xstuff::xc`, `xstuff::xcdot`, and `xstuff::xstore`.

Referenced by `vmec()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.54 src/interp.f90 File Reference

Interpolate R , Z and λ on full grid.

Functions/Subroutines

- subroutine `interp` (`xnew`, `xold`, `scalxc`, `nsnew`, `nsold`)
Interpolate R , Z and λ on full grid.

7.54.1 Detailed Description

Interpolate R , Z and λ on full grid.

7.54.2 Function/Subroutine Documentation

7.54.2.1 `interp()`

```
subroutine interp (
    real(rprec), dimension(nsnew,mnsize,3*ntmax), intent(out) xnew,
    real(rprec), dimension(nsold,mnsize,3*ntmax), intent(inout) xold,
    real(rprec), dimension(nsnew,mnsize,3*ntmax), intent(in) scalxc,
    integer, intent(in) nsnew,
    integer, intent(in) nsold )
```

Interpolate R , Z and λ on full grid.

Parameters

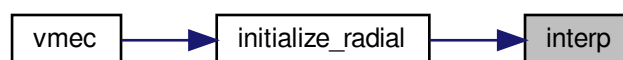
<i>xnew</i>	interpolated state vector (nsnew surfaces)
<i>xold</i>	interpolation basis: old state vector (nsold surfaces)
<i>scalxc</i>	scaling factors to normalize the new state vector to
<i>nsnew</i>	new number of flux surfaces
<i>nsold</i>	old number of flux surfaces

Definition at line 11 of file `interp.f90`.

References `vmec_persistent::ixm`, and `vmec_params::ntmax`.

Referenced by `initialize_radial()`.

Here is the caller graph for this function:



7.55 src/jacobian.f90 File Reference

Evaluate the Jacobian of the transform from flux- to cylindrical coordinates.

Functions/Subroutines

- subroutine [jacobian](#)
Evaluate the Jacobian of the transform from flux- to cylindrical coordinates.

7.55.1 Detailed Description

Evaluate the Jacobian of the transform from flux- to cylindrical coordinates.

7.55.2 Function/Subroutine Documentation

7.55.2.1 jacobian()

```
subroutine jacobian
```

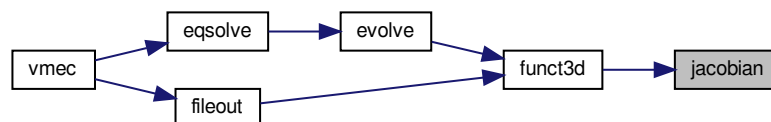
Evaluate the Jacobian of the transform from flux- to cylindrical coordinates.

Definition at line 6 of file jacobian.f90.

References `vforces::armn_e`, `vforces::armn_o`, `vforces::azmn_e`, `vforces::azmn_o`, `vforces::brmn_e`, `vforces::bzmn_e`, `vmec_main::irst`, `vmec_params::meven`, `vmec_params::modd`, `vmec_dim::ns`, `vmec_main::ohs`, `realspace::r1`, `realspace::ru`, `realspace::shalf`, `realspace::z1`, and `realspace::zu`.

Referenced by `funct3d()`.

Here is the caller graph for this function:



7.56 src/jxbforce.f90 File Reference

Program for computing local $\mathbf{K} \times \mathbf{B} = \nabla p$ force balance.

Functions/Subroutines

- subroutine [jxbforce](#) (bsupu, bsupv, bsubu, bsubv, bsubsh, bsubsu, bsubsv, gsqrt, bsq, itheta, izeta, brho, ier_flag)

Program for computing local $\mathbf{K} \times \mathbf{B} = \nabla p$ force balance.

7.56.1 Detailed Description

Program for computing local $\mathbf{K} \times \mathbf{B} = \nabla p$ force balance.

7.56.2 Function/Subroutine Documentation

7.56.2.1 jxbforce()

```
subroutine jxbforce (
    real(rprec), dimension(ns,nznt), intent(in) bsupu,
    real(rprec), dimension(ns,nznt), intent(in) bsupv,
    real(rprec), dimension(ns,nznt,0:1), intent(inout), target bsubu,
    real(rprec), dimension(ns,nznt,0:1), intent(inout), target bsubv,
    real(rprec), dimension(ns,nznt), intent(in) bsubsh,
    real(rprec), dimension(ns,nznt,0:1) bsubsu,
    real(rprec), dimension(ns,nznt,0:1) bsubsv,
    real(rprec), dimension(ns,nznt), intent(in) gsqrt,
    real(rprec), dimension(ns,nznt), intent(in) bsq,
    real(rprec), dimension(ns,nznt), intent(out) itheta,
    real(rprec), dimension(ns,nznt), intent(out) izeta,
    real(rprec), dimension(ns,nznt), intent(out) brho,
    integer, intent(in) ier_flag )
```

Program for computing local $\mathbf{K} \times \mathbf{B} = \nabla p$ force balance.

Parameters

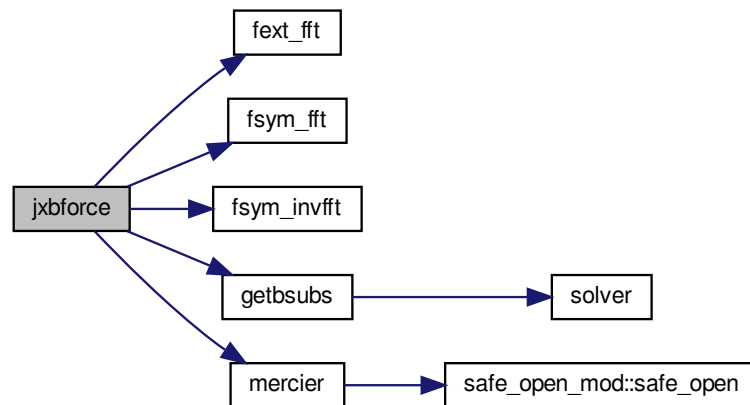
<i>bsupu</i>	contravariant component of magnetic field B^θ
<i>bsupv</i>	contravariant component of magnetic field B^ζ
<i>bsubu</i>	covariant component of magnetic field B_θ
<i>bsubv</i>	covariant component of magnetic field B_ζ
<i>bsubsh</i>	covariant component of magnetic field B_s (on half grid?)
<i>bsubsu</i>	tangential derivate of covariant component of magnetic field $\partial B_s / \partial \theta$ (?)
<i>bsubsv</i>	tangential derivate of covariant component of magnetic field $\partial B_s / \partial \zeta$ (?)
<i>gsqrt</i>	Jacobian \sqrt{g}
<i>bsq</i>	modulus of magnetic field $ \mathbf{B} ^2$
<i>itheta</i>	index in poloidal direction
<i>izeta</i>	index in toroidal direction
<i>brho</i>	radial component of magnetic field B_ρ (?)
<i>ier_flag</i>	error flag

Definition at line 19 of file jxbforce.f90.

References vmec_main::bdotb, vmec_main::bdotgradv, fext_fft(), fsym_fft(), fsym_invfft(), getbsubs(), realspace::guu, realspace::guv, realspace::gvv, vmec_main::iotas, vmec_main::jdotb, vmec_main::jpar2, vmec_main::jperp2, mercier(), vmec_params::mnyq, vmec_params::nnyq, vmec_main::ohs, vmec_main::phi, realspace::phip, vmec_main::phips, vmec_main::pres, vmec_main::r0scale, realspace::r1, realspace::ru, realspace::rv, realspace::shalf, vmec_params::signgs, vmec_params::successful_term_flag, vmec_main::vp, realspace::wint, realspace::zu, and realspace::zv.

Referenced by eqfor().

Here is the call graph for this function:



Here is the caller graph for this function:



7.57 src/lamcal.f90 File Reference

Normalization parameters for λ .

Functions/Subroutines

- subroutine `lamcal` (overg, guu, guv, gvv)
Normalization parameters for λ .

7.57.1 Detailed Description

Normalization parameters for λ .

7.57.2 Function/Subroutine Documentation

7.57.2.1 lamcal()

```
subroutine lamcal (
    real(rprec), dimension(ns,nznt), intent(in) overg,
    real(rprec), dimension(ns,nznt), intent(in) guu,
    real(rprec), dimension(ns,nznt), intent(in) guv,
    real(rprec), dimension(ns,nznt), intent(in) gvv )
```

Normalization parameters for λ .

Parameters

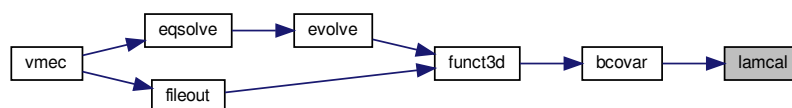
<i>overg</i>	inverse of Jacobian $1/\sqrt{g}$
<i>guu</i>	metric element $g_{\theta\theta}$
<i>guv</i>	metric element $g_{\theta\zeta}$
<i>gvv</i>	metric element $g_{\zeta\zeta}$

Definition at line 10 of file lamcal.f90.

References vmec_main::blam, vmec_main::clam, vmec_main::dlam, vmec_main::faclam, vmec_params::jlam, vmec_params::lamscale, vmec_params::ntmax, vmec_main::r0scale, and realspace::sqrts.

Referenced by bcovar().

Here is the caller graph for this function:



7.58 src/line_segment.f File Reference

This module contains code to create a profile constructed of line segments.

Modules

- module [line_segment](#)

This module contains code to create a profile constructed of line segments. These line segments are assumed to be specified such that $xx(i) < xx(i + 1)$.

Functions/Subroutines

- subroutine, public [line_segment::line_seg](#) (x, y, xx, yy, n)
- subroutine, public [line_segment::line_seg_int](#) (x, y, xx, yy, n)
- logical function, public [line_segment::line_seg_test](#) ()

7.58.1 Detailed Description

This module contains code to create a profile constructed of line segments.

7.59 src/magnetic_fluxes.f90 File Reference

Compute toroidal and poloidal magnetic flux profiles.

Functions/Subroutines

- real(rprec) function [torflux_deriv](#) (x)
Compute the radial derivative of the enclosed toroidal magnetic flux.
- real(rprec) function [polflux_deriv](#) (x)
Compute the radial derivative of the enclosed poloidal magnetic flux.
- real(rprec) function [torflux](#) (x)
Compute the enclosed toroidal magnetic flux.
- real(rprec) function [polflux](#) (x)
Compute the enclosed poloidal magnetic flux.

7.59.1 Detailed Description

Compute toroidal and poloidal magnetic flux profiles.

7.59.2 Function/Subroutine Documentation

7.59.2.1 polflux()

```
real(rprec) function polflux (
    real(rprec), intent(in) x )
```

Compute the enclosed poloidal magnetic flux.

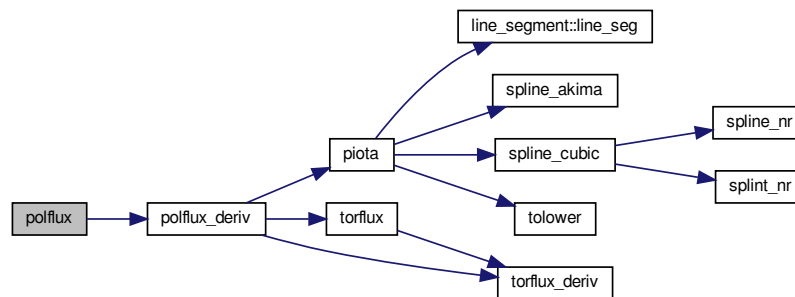
Parameters

	<i>x</i>	evaluation location
<i>in</i>	<i>x</i>	radial flux variable (=TOROIDAL FLUX ONLY IF APhi=1)

Definition at line 75 of file magnetic_fluxes.f90.

References polflux_deriv().

Here is the call graph for this function:



7.59.2.2 polflux_deriv()

```

real(rprec) function polflux_deriv (
    real(rprec), intent(in) x )

```

Compute the radial derivative of the enclosed poloidal magnetic flux.

Parameters

	<i>x</i>	evaluation location
<i>in</i>	<i>x</i>	radial flux variable (=TOROIDAL FLUX ONLY IF APhi=1)

Returns

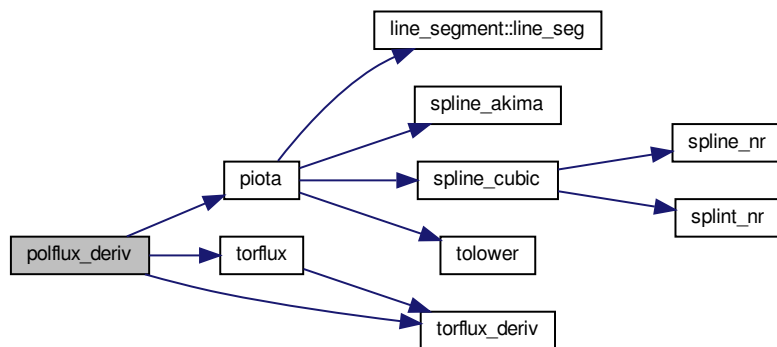
`polflux_deriv == d(chi)/dx = iota(TF(x)) * torflux_deriv(x)`

Definition at line 28 of file magnetic_fluxes.f90.

References `piota()`, `torflux()`, and `torflux_deriv()`.

Referenced by `polflux()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.59.2.3 torflux()

```

real(rprec) function torflux (
    real(rprec), intent(in) x )

```

Compute the enclosed toroidal magnetic flux.

Parameters

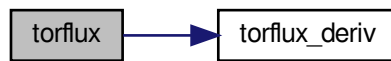
	<i>x</i>	evaluation location
in	<i>x</i>	radial flux variable (=TOROIDAL FLUX ONLY IF APhi=1)

Definition at line 51 of file `magnetic_fluxes.f90`.

References `torflux_deriv()`.

Referenced by `polflux_deriv()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.59.2.4 torflux_deriv()

```

real(rprec) function torflux_deriv (
    real(rprec), intent(in) x )
  
```

Compute the radial derivative of the enclosed toroidal magnetic flux.

Parameters

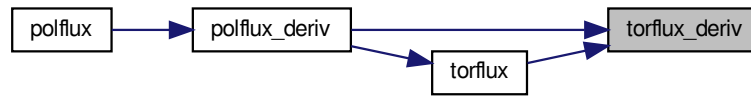
	<i>x</i>	evaluation location
in	<i>x</i>	radial flux variable (=TOROIDAL FLUX ONLY IF APhi=1)

Definition at line 7 of file magnetic_fluxes.f90.

References vmec_input::aphi.

Referenced by polflux_deriv(), and torflux().

Here is the caller graph for this function:



7.60 src/mercier.f90 File Reference

Evaluate the Mercier stability criterion.

Functions/Subroutines

- subroutine [mercier](#) (gsqrt, bsq, bdotj, iotas, wint, r1, rt, rz, zt, zz, bsubu, vp, phips, pres, ns, nznt)
Evaluate the Mercier stability criterion.

7.60.1 Detailed Description

Evaluate the Mercier stability criterion.

7.60.2 Function/Subroutine Documentation

7.60.2.1 mercier()

```

subroutine mercier (
    real(rprec), dimension(ns,nznt), intent(in) gsqrt,
    real(rprec), dimension(ns,nznt), intent(in) bsq,
    real(rprec), dimension(ns,nznt), intent(inout) bdotj,
    real(rprec), dimension(ns), intent(in) iotas,
    real(rprec), dimension(ns*nznt), intent(in) wint,
    real(rprec), dimension(ns,nznt,0:1), intent(in) r1,
    real(rprec), dimension(ns,nznt,0:1), intent(in) rt,
    real(rprec), dimension(ns,nznt,0:1), intent(in) rz,
    real(rprec), dimension(ns,nznt,0:1), intent(in) zt,
    real(rprec), dimension(ns,nznt,0:1), intent(in) zz,
    real(rprec), dimension(ns*nznt), intent(in) bsubu,
    real(rprec), dimension(ns), intent(in) vp,
    real(rprec), dimension(ns), intent(in) phips,
    real(rprec), dimension(ns), intent(in) pres,
    integer, intent(in) ns,
    integer, intent(in) nznt )

```

Evaluate the Mercier stability criterion.

Parameters

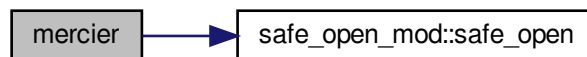
<i>gsqrt</i>	Jacobian \sqrt{g}
<i>bsq</i>	modulus of magnetic field $ \mathbf{B} $
<i>bdotj</i>	parallel current density $\mathbf{B} \cdot \mathbf{j}$
<i>iotas</i>	rotational transform profile
<i>wint</i>	normalization constant for flux-surface integrals
<i>r1</i>	R
<i>rt</i>	$\partial R / \partial \theta$
<i>rz</i>	$\partial R / \partial \zeta$
<i>zt</i>	$\partial Z / \partial \theta$
<i>zz</i>	$\partial Z / \partial \zeta$
<i>bsubu</i>	contravariant component of magnetic field B^ζ
<i>vp</i>	radial profile of specific volume $\partial V / \partial s$
<i>phips</i>	radial derivative of enclosed toroidal magnetic flux
<i>pres</i>	pressure profile
<i>ns</i>	number of flux surfaces
<i>nznt</i>	number of grid points per flux surface

Definition at line 22 of file mercier.f90.

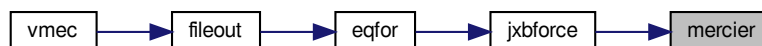
References vmecier::dcurr, vmecier::dgeod, vmecier::dmerc, stel_kinds::dp, vmecier::dshear, vmecier::dwell, vmec_input::input_extension, vparams::nmercier0, and safe_open_mod::safe_open().

Referenced by jxbforce().

Here is the call graph for this function:



Here is the caller graph for this function:



7.61 src/mgrid_mod.f File Reference

Precomputed table of magnetic field due to confinent coils.

Modules

- module [mgrid_mod](#)

Precomputed table of magnetic field due to confinement coils.

Functions/Subroutines

- subroutine [mgrid_mod::read_mgrid](#) (mgrid_file, extcur, nv, nfp, lscreen, ier_flag)
- subroutine [mgrid_mod::sum_bfield](#) (bfield, bf_add, cur, n1)
- subroutine [mgrid_mod::assign_bptrs](#) (bptr)
- subroutine [mgrid_mod::free_mgrid](#) (istat)

Variables

- integer, parameter [mgrid_mod::nlimset](#) = 2
- character(len= *), parameter [mgrid_mod::vn_br0](#) = 'br'
- character(len= *), parameter [mgrid_mod::vn_bp0](#) = 'bp'
- character(len= *), parameter [mgrid_mod::vn_bz0](#) = 'bz'
- character(len= *), parameter [mgrid_mod::vn_ir](#) = 'ir'
- character(len= *), parameter [mgrid_mod::vn_jz](#) = 'jz'
- character(len= *), parameter [mgrid_mod::vn_kp](#) = 'kp'
- character(len= *), parameter [mgrid_mod::vn_nfp](#) = 'nfp'
- character(len= *), parameter [mgrid_mod::vn_rmin](#) = 'rmin'
- character(len= *), parameter [mgrid_mod::vn_rmax](#) = 'rmax'
- character(len= *), parameter [mgrid_mod::vn_zmin](#) = 'zmin'
- character(len= *), parameter [mgrid_mod::vn_zmax](#) = 'zmax'
- character(len= *), parameter [mgrid_mod::vn_coilgrp](#) = 'coil_group'
- character(len= *), parameter [mgrid_mod::vn_nextcur](#) = 'nextcur'
- character(len= *), parameter [mgrid_mod::vn_mgmode](#) = 'mgrid_mode'
- character(len= *), parameter [mgrid_mod::vn_coilcur](#) = 'raw_coil_cur'
- character(len= *), parameter [mgrid_mod::ln_next](#) = 'External currents'
- integer [mgrid_mod::nr0b](#)
- integer [mgrid_mod::np0b](#)
- integer [mgrid_mod::nfper0](#)
- integer [mgrid_mod::nz0b](#)
- integer [mgrid_mod::nobd](#)
- integer [mgrid_mod::nobser](#)
- integer [mgrid_mod::nextcur](#)
- integer [mgrid_mod::nbfldn](#)
- integer [mgrid_mod::nbsets](#)
- integer [mgrid_mod::nbcoilsn](#)
- integer [mgrid_mod::nbvac](#)
- integer [mgrid_mod::nbcoil_max](#)
- integer [mgrid_mod::nlim](#)
- integer [mgrid_mod::nlim_max](#)
- integer [mgrid_mod::nsets](#)
- integer [mgrid_mod::nrgrid](#)
- integer [mgrid_mod::nzgrid](#)
- integer, dimension(:), allocatable [mgrid_mod::needflx](#)
- integer, dimension(:), allocatable [mgrid_mod::nbcoils](#)
- integer, dimension(:), allocatable [mgrid_mod::limitr](#)
- integer, dimension(:), allocatable [mgrid_mod::nsetsn](#)
- integer, dimension(:, :), allocatable [mgrid_mod::iconnect](#)

- integer, dimension(:,:), allocatable [mgrid_mod::needbfld](#)
- real(rprec) [mgrid_mod::rminb](#)
- real(rprec) [mgrid_mod::zminb](#)
- real(rprec) [mgrid_mod::rmaxb](#)
- real(rprec) [mgrid_mod::zmaxb](#)
- real(rprec) [mgrid_mod::delrb](#)
- real(rprec) [mgrid_mod::delzb](#)
- real(rprec) [mgrid_mod::rx1](#)
- real(rprec) [mgrid_mod::rx2](#)
- real(rprec) [mgrid_mod::zy1](#)
- real(rprec) [mgrid_mod::zy2](#)
- real(rprec) [mgrid_mod::condif](#)
- real(rprec), dimension(:,:), allocatable, target [mgrid_mod::bvac](#)
- real(rprec), dimension(:,:), pointer [mgrid_mod::brvac](#)
- real(rprec), dimension(:,:), pointer [mgrid_mod::bzbvac](#)
- real(rprec), dimension(:,:), pointer [mgrid_mod::bpvac](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::unpsiext](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::plbfld](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::rbcoil](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::zbcoil](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::abcoil](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::bcoil](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::rbcoilsqr](#)
- real(rprec), dimension(:), allocatable [mgrid_mod::raw_coil_current](#)
- real(rprec), dimension(:), allocatable [mgrid_mod::xobser](#)
- real(rprec), dimension(:), allocatable [mgrid_mod::zobser](#)
- real(rprec), dimension(:), allocatable [mgrid_mod::xobsqr](#)
- real(rprec), dimension(:), allocatable [mgrid_mod::dsiext](#)
- real(rprec), dimension(:), allocatable [mgrid_mod::psiext](#)
- real(rprec), dimension(:), allocatable [mgrid_mod::plflux](#)
- real(rprec), dimension(:), allocatable [mgrid_mod::b_chi](#)
- character(len=300) [mgrid_mod::mgrid_path](#)
- character(len=300) [mgrid_mod::mgrid_path_old](#) = " "
- character(len=30), dimension(:), allocatable [mgrid_mod::curlabel](#)
- character(len=15), dimension(:), allocatable [mgrid_mod::dsilabel](#)
- character(len=15), dimension(:), allocatable [mgrid_mod::bloopnames](#)
- character(len=30) [mgrid_mod::tokid](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::dbcoil](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::pfcspec](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::rlim](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::zlim](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::reslim](#)
- real(rprec), dimension(:,:), allocatable [mgrid_mod::seplim](#)
- character(len=1) [mgrid_mod::mgrid_mode](#)

7.61.1 Detailed Description

Precomputed table of magnetic field due to confinent coils.

7.62 src/NESTOR/analysum.f90 File Reference

Functions/Subroutines

- subroutine [analysum](#) (grpnm, bvec, sl, tl, m, n, l, ivacskip, lasym, m_map, n_map, grpnm_m_map, grpnm_n_map)

7.62.1 Function/Subroutine Documentation

7.62.1.1 analysum()

```

subroutine analysum (
    real(rprec), dimension(0:mf,-nf:nf,nuv2,ndim), intent(inout) grpnm,
    real(rprec), dimension(0:mf,-nf:nf,ndim), intent(inout) bvec,
    real(rprec), dimension(nuv2), intent(in) sl,
    real(rprec), dimension(nuv2), intent(in) tl,
    integer, intent(in) m,
    integer, intent(in) n,
    integer, intent(in) l,
    integer, intent(in) ivacskip,
    logical, intent(in) lasym,
    real(rprec), dimension(0:mf,-nf:nf), intent(inout) m_map,
    real(rprec), dimension(0:mf,-nf:nf), intent(inout) n_map,
    real(rprec), dimension(0:mf,-nf:nf,nuv2), intent(inout) grpnm_m_map,
    real(rprec), dimension(0:mf,-nf:nf,nuv2), intent(inout) grpnm_n_map )

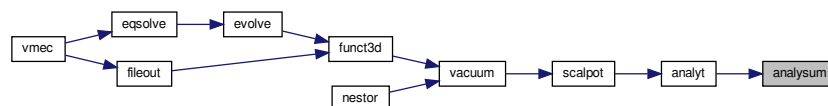
```

Definition at line 2 of file analysum.f90.

References `vacmod::bexni`, and `vacmod::grpnm`.

Referenced by `analyt()`.

Here is the caller graph for this function:



7.63 src/NESTOR/analysum2.f90 File Reference

Functions/Subroutines

- subroutine [analysum2](#) (grpnm, bvec, m, n, l, ivacskip, lasym, m_map, n_map, grpnm_m_map, grpnm_n_map)

7.63.1 Function/Subroutine Documentation

7.63.1.1 analysum2()

```

subroutine analysum2 (
    real(rprec), dimension(0:mf,-nf:nf,nuv2,ndim), intent(inout) grpmn,
    real(rprec), dimension(0:mf,-nf:nf,ndim), intent(inout) bvec,
    integer, intent(in) m,
    integer, intent(in) n,
    integer, intent(in) l,
    integer, intent(in) ivacskip,
    logical, intent(in) lasym,
    real(rprec), dimension(0:mf,-nf:nf), intent(inout) m_map,
    real(rprec), dimension(0:mf,-nf:nf), intent(inout) n_map,
    real(rprec), dimension(0:mf,-nf:nf,nuv2), intent(inout) grpmn_m_map,
    real(rprec), dimension(0:mf,-nf:nf,nuv2), intent(inout) grpmn_n_map )

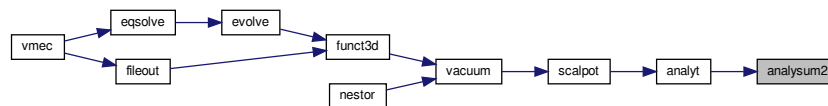
```

Definition at line 2 of file analysum2.f90.

References vacmod::bexni, vacmod::grpmn, vacmod::slm, vacmod::slp, vacmod::tIm, and vacmod::tIp.

Referenced by analyt().

Here is the caller graph for this function:



7.64 src/NESTOR/analyt.f90 File Reference

Functions/Subroutines

- subroutine [analyt](#) (grpmn, bvec, ivacskip, lasym, m_map, n_map, grpmn_m_map, grpmn_n_map)

7.64.1 Function/Subroutine Documentation

7.64.1.1 `analyt()`

```

subroutine analyt (
    real(rprec), dimension(nuv2*mnpd2), intent(out) grpmn,
    real(rprec), dimension(mnpd2), intent(out) bvec,
    integer, intent(in) ivacskip,
    integer, intent(in) lasym,
    real(rprec), dimension(mnpd2), intent(out) m_map,
    real(rprec), dimension(mnpd2), intent(out) n_map,
    real(rprec), dimension(nuv2*mnpd2), intent(out) grpmn_m_map,
    real(rprec), dimension(nuv2*mnpd2), intent(out) grpmn_n_map )

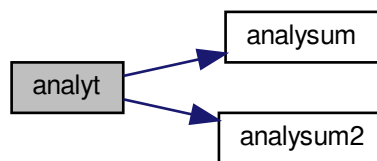
```

Definition at line 2 of file `analyt.f90`.

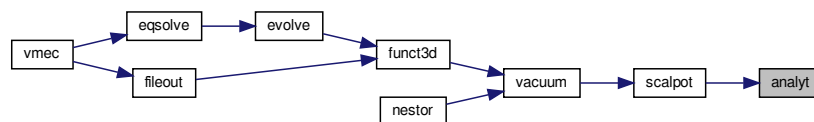
References `vacmod::adm`, `vacmod::adp`, `vacmod::all_slm`, `vacmod::all_slp`, `vacmod::all_tlm`, `vacmod::all_tlp`, `analysum()`, `analysum2()`, `vacmod::auu`, `vacmod::auv`, `vacmod::avv`, `vacmod::azm1u`, `vacmod::azp1u`, `vacmod::cma`, `vacmod::cma11u`, `vacmod::delt1u`, `vacmod::grpmn`, `vacmod::guu_b`, `vacmod::guv_b`, `vacmod::gvv_b`, `vacmod::r0m`, `vacmod::r0p`, `vacmod::r1m`, `vacmod::r1p`, `vacmod::ra1m`, `vacmod::ra1p`, `vacmod::slm`, `vacmod::slp`, `vacmod::slpm`, `vacmod::sqad1u`, `vacmod::sqad2u`, `vacmod::sqrtc`, `vacmod::t1m`, `vacmod::t1m1`, `vacmod::t1m2`, `vacmod::tlp`, `vacmod::tlp1`, `vacmod::tlp2`, `vacmod::tlpm`, and `vacmod::two`.

Referenced by `scalpot()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.65 `src/NESTOR/becoil.f90` File Reference

Functions/Subroutines

- subroutine `becoil` (`rad`, `zee`, `brvac`, `bpvac`, `bzvac`)

7.65.1 Function/Subroutine Documentation

7.65.1.1 becoil()

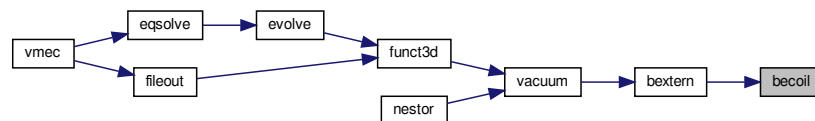
```
subroutine becoil (
    real(rprec), dimension(nuv2), intent(in) rad,
    real(rprec), dimension(nuv2), intent(in) zee,
    real(rprec), dimension(nr0b,nz0b,np0b), intent(in) brvac,
    real(rprec), dimension(nr0b,nz0b,np0b), intent(in) bpvac,
    real(rprec), dimension(nr0b,nz0b,np0b), intent(in) bzvac )
```

Definition at line 2 of file becoil.f90.

References `vacmod::bphi`, `vacmod::brad`, `vacmod::bz`, `mgrid_mod::delrb`, `mgrid_mod::delzb`, `mgrid_mod::np0b`, `mgrid_mod::nr0b`, `mgrid_mod::nz0b`, `mgrid_mod::rmaxb`, `mgrid_mod::rminb`, `mgrid_mod::zmaxb`, and `mgrid_mod::zminb`.

Referenced by `bextern()`.

Here is the caller graph for this function:



7.66 src/NESTOR/belicu.f90 File Reference

Functions/Subroutines

- subroutine `belicu` (`torcur`, `bx`, `by`, `bz`, `cos1`, `sin1`, `rp`, `zp`)

7.66.1 Function/Subroutine Documentation

7.66.1.1 belicu()

```

subroutine belicu (
    real(rprec), intent(in) torcur,
    real(rprec), dimension(nuv2), intent(out) bx,
    real(rprec), dimension(nuv2), intent(out) by,
    real(rprec), dimension(nuv2), intent(out) bz,
    real(rprec), dimension(nuv2), intent(in) cos1,
    real(rprec), dimension(nuv2), intent(in) sin1,
    real(rprec), dimension(nuv2), intent(in) rp,
    real(rprec), dimension(nuv2), intent(in) zp )

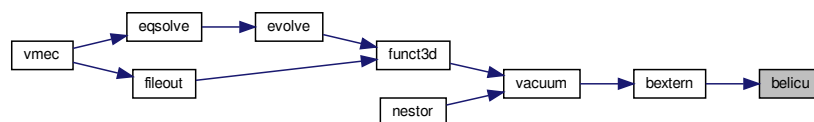
```

Definition at line 2 of file belicu.f90.

References vacmod::bz, vacmod::raxis_nestor, vacmod::xpts, and vacmod::zaxis_nestor.

Referenced by bextern().

Here is the caller graph for this function:



7.67 src/NESTOR/bextern.f90 File Reference

Functions/Subroutines

- subroutine [bextern](#) (plascur, wint)

7.67.1 Function/Subroutine Documentation

7.67.1.1 bextern()

```

subroutine bextern (
    real(rprec), intent(in) plascur,
    real(rprec), dimension(nuv2), intent(in) wint )

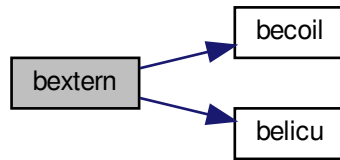
```

Definition at line 2 of file bextern.f90.

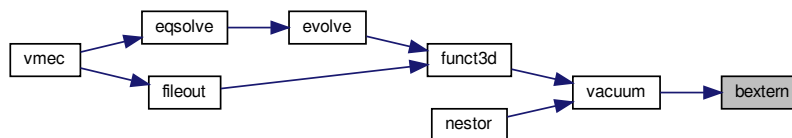
References becoil(), belicu(), vacmod::bexn, vacmod::bexni, vacmod::bexu, vacmod::bexv, vacmod::bphi, vacmod::brad, mgrid_mod::bvac, vacmod::bz, vacmod::pi2, vacmod::r1b, vacmod::rub, vacmod::rvb, vacmod::snr, vacmod::snv, vacmod::snz, vacmod::z1b, vacmod::zub, and vacmod::zvb.

Referenced by vacuum().

Here is the call graph for this function:



Here is the caller graph for this function:



7.68 src/NESTOR/data/nestor_io.f90 File Reference

Input and Output for stand-alone NESTOR.

Modules

- module [nestor_io](#)
Input and Output for stand-alone NESTOR.

Functions/Subroutines

- subroutine [nestor_io::read_nestor_inputs](#) (vac_file)
- subroutine [nestor_io::write_nestor_outputs](#) (vac_file, lasym, ivac, ier_flag)
- subroutine [write_nestor_inputs](#) (vac_file, vacuum_calls, ier_flag, mgrid_file, input_extension, ivacskip, ivac, nfp, ntor, mpol, nzeta, ntheta, mnmax, xm, xn, rmnc, zmns, rmns, zmnc, rbtor, ctor, lasym, signgs, extcur_↵ nestor, raxis_nestor, zaxis_nestor, wint, nznt, amatsav, bvecsav, mnpd2, bsubvvac)
- subroutine [read_nestor_outputs](#) (vac_file, ier_flag, ivac)

Variables

- character(len=255) [nestor_io::input_extension](#)
- character(len=255) [nestor_io::mgrid_file](#)
- real(dp), dimension(:), allocatable [nestor_io::extcur](#)
- real(dp), dimension(:), allocatable [nestor_io::raxis](#)
- real(dp), dimension(:), allocatable [nestor_io::zaxis](#)
- real(dp), dimension(:), allocatable [nestor_io::xm](#)
- real(dp), dimension(:), allocatable [nestor_io::xn](#)
- real(dp), dimension(:), allocatable [nestor_io::rmnc](#)
- real(dp), dimension(:), allocatable [nestor_io::zmns](#)
- real(dp), dimension(:), allocatable [nestor_io::rmns](#)
- real(dp), dimension(:), allocatable [nestor_io::zmnc](#)
- real(dp), dimension(:), allocatable [nestor_io::wint](#)
- integer [nestor_io::nfp](#)
- integer [nestor_io::ntor](#)
- integer [nestor_io::mpol](#)
- integer [nestor_io::ntheta](#)
- integer [nestor_io::nzeta](#)
- integer [nestor_io::nextcur](#)
- integer [nestor_io::ier_flag](#)
- integer [nestor_io::ivac](#)
- integer [nestor_io::ivacskip](#)
- integer [nestor_io::mnmax](#)
- integer [nestor_io::vacuum_calls](#)
- logical [nestor_io::lasym](#)
- real(dp) [nestor_io::ctor](#)
- real(dp) [nestor_io::rbtor](#)
- real(dp) [nestor_io::signgs](#)
- integer [nestor_io::mnpd2_nestor](#)
- real(dp), dimension(:), allocatable [nestor_io::amatsav_nestor](#)
- real(dp), dimension(:), allocatable [nestor_io::bvecsav_nestor](#)
- real(dp) [nestor_io::bsubvvac_nestor](#)
- character(len= *), dimension(1), parameter [nestor_io::mn1dim](#) = ('mn_mode'/)
- character(len= *), dimension(1), parameter [nestor_io::mnpotdim](#) = ('mn_mode_pot'/)
- character(len= *), dimension(1), parameter [nestor_io::nzntdim](#) = ('nznnt'/)
- character(len= *), dimension(1), parameter [nestor_io::nzetadim](#) = ('nzeta'/)
- character(len= *), dimension(1), parameter [nestor_io::nextcurim](#) = ('nextcur'/)
- character(len= *), dimension(1), parameter [nestor_io::bvecsavdim](#) = ('mnpd2'/)
- character(len= *), dimension(1), parameter [nestor_io::amatsavdim](#) = ('mnpd2_times_mnpd2'/)
- character(len= *), dimension(2), parameter [nestor_io::r2dim](#) = ('mn_mode','radius '/)
- character(len= *), parameter [nestor_io::vn_vacuum_calls](#) = 'vacuum_calls'
- character(len= *), parameter [nestor_io::vn_ier_flag](#) = "ier_flag"
- character(len= *), parameter [nestor_io::vn_mgrid](#) = "mgrid_file"
- character(len= *), parameter [nestor_io::vn_inputext](#) = "input_extension"
- character(len= *), parameter [nestor_io::vn_ivacskip](#) = "ivacskip"
- character(len= *), parameter [nestor_io::vn_ivac](#) = "ivac"
- character(len= *), parameter [nestor_io::vn_nfp](#) = "nfp"
- character(len= *), parameter [nestor_io::vn_ntor](#) = "ntor"
- character(len= *), parameter [nestor_io::vn_mpol](#) = "mpol"
- character(len= *), parameter [nestor_io::vn_nzeta](#) = "nzeta"
- character(len= *), parameter [nestor_io::vn_ntheta](#) = "ntheta"
- character(len= *), parameter [nestor_io::vn_mnmax](#) = "mnmax"
- character(len= *), parameter [nestor_io::vn_pmod](#) = "xm"
- character(len= *), parameter [nestor_io::vn_tmod](#) = "xn"

- character(len= *), parameter nestor_io::vn_rmnc = "rmnc"
- character(len= *), parameter nestor_io::vn_zmns = "zmns"
- character(len= *), parameter nestor_io::vn_rmns = "rmns"
- character(len= *), parameter nestor_io::vn_zmnc = "zmnc"
- character(len= *), parameter nestor_io::vn_rbtor = "rbtor"
- character(len= *), parameter nestor_io::vn_ctor = "ctor"
- character(len= *), parameter nestor_io::vn_lasym = "lasym"
- character(len= *), parameter nestor_io::vn_signgs = "signgs"
- character(len= *), parameter nestor_io::vn_extcur = "extcur"
- character(len= *), parameter nestor_io::vn_raxis_nestor = "raxis_nestor"
- character(len= *), parameter nestor_io::vn_zaxis_nestor = "zaxis_nestor"
- character(len= *), parameter nestor_io::vn_wint = "wint"
- character(len= *), parameter nestor_io::vn_bsqvac = "bsqvac"
- character(len= *), parameter nestor_io::vn_mnpd = "mnpd"
- character(len= *), parameter nestor_io::vn_xmpot = "xmpot"
- character(len= *), parameter nestor_io::vn_xnpot = "xnpot"
- character(len= *), parameter nestor_io::vn_potvac = "potvac"
- character(len= *), parameter nestor_io::vn_brv = "brv"
- character(len= *), parameter nestor_io::vn_bphiv = "bphiv"
- character(len= *), parameter nestor_io::vn_bzv = "bzv"
- character(len= *), parameter nestor_io::vn_bsubvvac = "bsubvvac"
- character(len= *), parameter nestor_io::vn_amatsav = "amatsav"
- character(len= *), parameter nestor_io::vn_bvecsav = "bvecsav"
- character(len= *), parameter nestor_io::vn_mnpd2 = "mnpd2"
- character(len= *), parameter nestor_io::vn_r1b = "r1b"
- character(len= *), parameter nestor_io::vn_rub = "rub"
- character(len= *), parameter nestor_io::vn_rvb = "rvb"
- character(len= *), parameter nestor_io::vn_z1b = "z1b"
- character(len= *), parameter nestor_io::vn_zub = "zub"
- character(len= *), parameter nestor_io::vn_zvb = "zvb"
- character(len= *), parameter nestor_io::vn_ruu = "ruu"
- character(len= *), parameter nestor_io::vn_ruv = "ruv"
- character(len= *), parameter nestor_io::vn_rvv = "rvv"
- character(len= *), parameter nestor_io::vn_zuu = "zuu"
- character(len= *), parameter nestor_io::vn_zuv = "zuv"
- character(len= *), parameter nestor_io::vn_zvv = "zvv"
- character(len= *), parameter nestor_io::vn_guu_b = "guu_b"
- character(len= *), parameter nestor_io::vn_guv_b = "guv_b"
- character(len= *), parameter nestor_io::vn_gvv_b = "gvv_b"
- character(len= *), parameter nestor_io::vn_rzb2 = "rzb2"
- character(len= *), parameter nestor_io::vn_snr = "snr"
- character(len= *), parameter nestor_io::vn_snv = "snv"
- character(len= *), parameter nestor_io::vn_snz = "snz"
- character(len= *), parameter nestor_io::vn_drv = "drv"
- character(len= *), parameter nestor_io::vn_auu = "auu"
- character(len= *), parameter nestor_io::vn_auv = "auv"
- character(len= *), parameter nestor_io::vn_avv = "avv"
- character(len= *), parameter nestor_io::vn_rcosuv = "rcosuv"
- character(len= *), parameter nestor_io::vn_rsinuv = "rsinuv"
- character(len= *), parameter nestor_io::vn_brad = "brad"
- character(len= *), parameter nestor_io::vn_bphi = "bphi"
- character(len= *), parameter nestor_io::vn_bz = "bz"
- character(len= *), parameter nestor_io::vn_bexu = "bexu"
- character(len= *), parameter nestor_io::vn_bexv = "bexv"
- character(len= *), parameter nestor_io::vn_bexn = "bexn"

- character(len= *), parameter `nestor_io::vn_bexni` = "bexni"
- character(len= *), parameter `nestor_io::vn_grpmn` = "grpmn"
- character(len= *), parameter `nestor_io::vn_adp` = "adp"
- character(len= *), parameter `nestor_io::vn_adm` = "adm"
- character(len= *), parameter `nestor_io::vn_cma` = "cma"
- character(len= *), parameter `nestor_io::vn_sqrtc` = "sqrtc"
- character(len= *), parameter `nestor_io::vn_sqrta` = "sqrta"
- character(len= *), parameter `nestor_io::vn_delt1u` = "delt1u"
- character(len= *), parameter `nestor_io::vn_azp1u` = "azp1u"
- character(len= *), parameter `nestor_io::vn_azm1u` = "azm1u"
- character(len= *), parameter `nestor_io::vn_cma11u` = "cma11u"
- character(len= *), parameter `nestor_io::vn_r1p` = "r1p"
- character(len= *), parameter `nestor_io::vn_r1m` = "r1m"
- character(len= *), parameter `nestor_io::vn_r0p` = "r0p"
- character(len= *), parameter `nestor_io::vn_r0m` = "r0m"
- character(len= *), parameter `nestor_io::vn_ra1p` = "ra1p"
- character(len= *), parameter `nestor_io::vn_ra1m` = "ra1m"
- character(len= *), parameter `nestor_io::vn_sqad1u` = "sqad1u"
- character(len= *), parameter `nestor_io::vn_sqad2u` = "sqad2u"
- character(len= *), parameter `nestor_io::vn_all_tlp` = "all_tlp"
- character(len= *), parameter `nestor_io::vn_all_tlm` = "all_tlm"
- character(len= *), parameter `nestor_io::vn_all_slp` = "all_slp"
- character(len= *), parameter `nestor_io::vn_all_slm` = "all_slm"
- character(len= *), parameter `nestor_io::vn_m_map` = "m_map"
- character(len= *), parameter `nestor_io::vn_n_map` = "n_map"
- character(len= *), parameter `nestor_io::vn_green` = "green"
- character(len= *), parameter `nestor_io::vn_greenp` = "greenp"
- character(len= *), parameter `nestor_io::vn_tanu` = "tanu"
- character(len= *), parameter `nestor_io::vn_tanv` = "tanv"
- character(len= *), parameter `nestor_io::vn_gstore` = "gstore"
- character(len= *), parameter `nestor_io::vn_grpmn_m_map` = "grpmn_m_map"
- character(len= *), parameter `nestor_io::vn_grpmn_n_map` = "grpmn_n_map"
- character(len= *), parameter `nestor_io::vn_imirr` = "imirr"
- character(len= *), parameter `nestor_io::vn_amatrix` = "amatrix"
- character(len= *), parameter `nestor_io::vn_potu` = "potu"
- character(len= *), parameter `nestor_io::vn_potv` = "potv"
- character(len= *), parameter `nestor_io::vn_bsubu` = "bsubu"
- character(len= *), parameter `nestor_io::vn_bsubv` = "bsubv"

7.68.1 Detailed Description

Input and Output for stand-alone NESTOR.

7.68.2 Function/Subroutine Documentation

7.68.2.1 read_nestor_outputs()

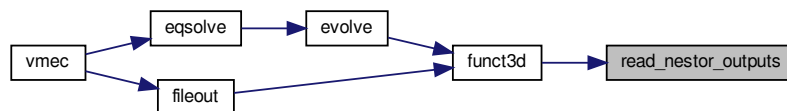
```
subroutine read_nestor_outputs (
    character(len=*), intent(in) vac_file,
    integer, intent(out) ier_flag,
    integer, intent(out) ivac )
```

Definition at line 633 of file nestor_io.f90.

References vacmod::amatsav, vacmod::bphiv, vacmod::brv, vacmod::bsqvac, vacmod::bsubvvac, vacmod::bvecsav, vacmod::bzv, stel_kinds::dp, vacmod::potvac, nestor_io::vn_amatsav, nestor_io::vn_bphiv, nestor_io::vn_brv, nestor_io::vn_bsqvac, nestor_io::vn_bsubvvac, nestor_io::vn_bvecsav, nestor_io::vn_bzv, nestor_io::vn_ier_flag, nestor_io::vn_ivac, nestor_io::vn_mnpd, nestor_io::vn_mnpd2, nestor_io::vn_potvac, nestor_io::vn_xmpot, and nestor_io::vn_xnpot.

Referenced by funct3d().

Here is the caller graph for this function:



7.68.2.2 write_nestor_inputs()

```
subroutine write_nestor_inputs (
    character(len=*), intent(in) vac_file,
    integer, intent(in) vacuum_calls,
    integer, intent(in) ier_flag,
    character(len=*), intent(in) mgrid_file,
    character(len=*), intent(in) input_extension,
    integer, intent(in) ivacskip,
    integer, intent(in) ivac,
    integer, intent(in) nfp,
    integer, intent(in) ntor,
    integer, intent(in) mpol,
    integer, intent(in) nzeta,
    integer, intent(in) ntheta,
    integer, intent(in) mnmax,
    real(dp), dimension(mnmax), intent(in) xm,
    real(dp), dimension(mnmax), intent(in) xn,
    real(dp), dimension(mnmax), intent(in) rmnc,
    real(dp), dimension(mnmax), intent(in) zmns,
    real(dp), dimension(mnmax), intent(in) rmns,
    real(dp), dimension(mnmax), intent(in) zmnc,
    real(dp), intent(in) rbtor,
    real(dp), intent(in) ctors)
```

```

logical, intent(in) lasym,
real(dp), intent(in) signgs,
real(dp), dimension(nextcur), intent(in) extcur_nestor,
real(dp), dimension(nzeta), intent(in) raxis_nestor,
real(dp), dimension(nzeta), intent(in) zaxis_nestor,
real(dp), dimension(nznt), intent(in) wint,
integer, intent(in) nznt,
real(dp), dimension(mnpd2*mnpd2), intent(in) amatsav,
real(dp), dimension(mnpd2), intent(in) bvecsav,
integer, intent(in) mnpd2,
real(dp), intent(in) bsubvvac )

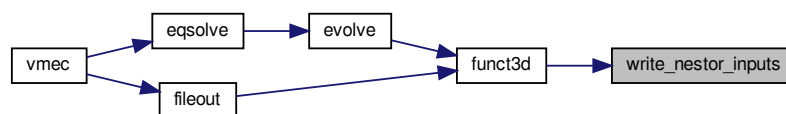
```

Definition at line 497 of file nestor_io.f90.

References nestor_io::amatsavdim, nestor_io::bvecsavdim, stel_kinds::dp, nestor_io::mn1dim, nestor_io::mnpotdim, mgrid_mod::nextcur, nestor_io::nextcurim, nestor_io::nzetadim, nestor_io::nzntdim, nestor_io::r2dim, nestor_io::vn_amatsav, nestor_io::vn_bsubvvac, nestor_io::vn_bvecsav, nestor_io::vn_ctor, nestor_io::vn_extcur, nestor_io::vn_ier_flag, nestor_io::vn_inputext, nestor_io::vn_ivac, nestor_io::vn_ivacskip, nestor_io::vn_lasym, nestor_io::vn_mgrid, nestor_io::vn_mnmax, nestor_io::vn_mnpd2, nestor_io::vn_mpol, nestor_io::vn_nfp, nestor_io::vn_ntheta, nestor_io::vn_ntor, nestor_io::vn_nzeta, nestor_io::vn_pmod, nestor_io::vn_raxis_nestor, nestor_io::vn_rbtor, nestor_io::vn_rmnc, nestor_io::vn_rmns, nestor_io::vn_signgs, nestor_io::vn_tmod, nestor_io::vn_vacuum_calls, nestor_io::vn_wint, nestor_io::vn_zaxis_nestor, nestor_io::vn_zmnc, and nestor_io::vn_zmns.

Referenced by funct3d().

Here is the caller graph for this function:



7.69 src/NESTOR/data/vac_persistent.f90 File Reference

Modules

- module [vac_persistent](#)

Variables

- integer, dimension(:), allocatable [vac_persistent::imirr](#)
- real(rprec), dimension(:), allocatable [vac_persistent::sinper](#)
- real(rprec), dimension(:), allocatable [vac_persistent::cosper](#)
- real(rprec), dimension(:), allocatable [vac_persistent::sinuv](#)
- real(rprec), dimension(:), allocatable [vac_persistent::cosuv](#)
- real(rprec), dimension(:), allocatable [vac_persistent::tanu](#)
- real(rprec), dimension(:), allocatable [vac_persistent::tanv](#)
- real(rprec), dimension(:), allocatable [vac_persistent::tanu_1d](#)
- real(rprec), dimension(:), allocatable [vac_persistent::tanv_1d](#)

- real(rprec), dimension(:), allocatable [vac_persistent::xmpot](#)
- real(rprec), dimension(:), allocatable [vac_persistent::xnpot](#)
- real(rprec), dimension(:), allocatable [vac_persistent::csign](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::sinu](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::cosu](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::sinv](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::cosv](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::sinui](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::cosui](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::sinu1](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::cosu1](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::sinv1](#)
- real(rprec), dimension(:,), allocatable [vac_persistent::cosv1](#)
- real(rprec), dimension(:,,:), allocatable [vac_persistent::cmns](#)
- real(rprec), dimension(:), allocatable [vac_persistent::bsubu_sur](#)
- real(rprec), dimension(:), allocatable [vac_persistent::bsubv_sur](#)
- real(rprec), dimension(:), allocatable [vac_persistent::bsupu_sur](#)
- real(rprec), dimension(:), allocatable [vac_persistent::bsupv_sur](#)

7.70 src/NESTOR/data/vacmod.f90 File Reference

Modules

- module [vacmod](#)

Functions/Subroutines

- subroutine [vacmod::allocate_nestor](#)
- subroutine [vacmod::free_mem_nestor](#)

Variables

- real(rprec), parameter [vacmod::p5](#) = cp5
- real(rprec), parameter [vacmod::two](#) = c2p0
- real(rprec) [vacmod::bsubvvac](#)
- real(rprec) [vacmod::pi2](#)
- real(rprec) [vacmod::pi3](#)
- real(rprec) [vacmod::pi4](#)
- real(rprec) [vacmod::alp](#)
- real(rprec) [vacmod::alu](#)
- real(rprec) [vacmod::alv](#)
- real(rprec) [vacmod::alvp](#)
- real(rprec) [vacmod::onp](#)
- real(rprec) [vacmod::onp2](#)
- logical [vacmod::precal_done](#)
- real(rprec), dimension(:), allocatable, target [vacmod::potvac](#)
- real(rprec), dimension(:), allocatable [vacmod::m_map_wrt](#)
- real(rprec), dimension(:), allocatable [vacmod::n_map_wrt](#)
- real(rprec), dimension(:), allocatable [vacmod::bvecsav](#)
- real(rprec), dimension(:), allocatable [vacmod::amatsav](#)

- `real(rprec), dimension(:), allocatable` [vacmod::bexni](#)
- `real(rprec), dimension(:), allocatable` [vacmod::brv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bphiv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bzbv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bsqvac](#)
- `real(rprec), dimension(:), allocatable` [vacmod::r1b](#)
- `real(rprec), dimension(:), allocatable` [vacmod::rub](#)
- `real(rprec), dimension(:), allocatable` [vacmod::rvb](#)
- `real(rprec), dimension(:), allocatable` [vacmod::z1b](#)
- `real(rprec), dimension(:), allocatable` [vacmod::zub](#)
- `real(rprec), dimension(:), allocatable` [vacmod::zvb](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bexu](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bexv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bexn](#)
- `real(rprec), dimension(:), allocatable` [vacmod::auu](#)
- `real(rprec), dimension(:), allocatable` [vacmod::auv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::avv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::snr](#)
- `real(rprec), dimension(:), allocatable` [vacmod::snv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::snz](#)
- `real(rprec), dimension(:), allocatable` [vacmod::drv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::guu_b](#)
- `real(rprec), dimension(:), allocatable` [vacmod::guv_b](#)
- `real(rprec), dimension(:), allocatable` [vacmod::gvv_b](#)
- `real(rprec), dimension(:), allocatable` [vacmod::rzb2](#)
- `real(rprec), dimension(:), allocatable` [vacmod::rcosuv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::rsinuv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::raxis_nestor](#)
- `real(rprec), dimension(:), allocatable` [vacmod::zaxis_nestor](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bsubu](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bsubv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::potu](#)
- `real(rprec), dimension(:), allocatable` [vacmod::potv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::amatrix](#)
- `real(rprec), dimension(:), allocatable` [vacmod::ruu](#)
- `real(rprec), dimension(:), allocatable` [vacmod::ruv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::rvv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::zuu](#)
- `real(rprec), dimension(:), allocatable` [vacmod::zuv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::zvv](#)
- `real(rprec), dimension(:), allocatable` [vacmod::brad](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bphi](#)
- `real(rprec), dimension(:), allocatable` [vacmod::bz](#)
- `real(rprec), dimension(:, :), allocatable` [vacmod::xpts](#)
- `real(rprec), dimension(:), allocatable` [vacmod::grpmn](#)
- `real(rprec), dimension(:), allocatable` [vacmod::grpmn_m_map_wrt](#)
- `real(rprec), dimension(:), allocatable` [vacmod::grpmn_n_map_wrt](#)
- `real(rprec), dimension(:), allocatable` [vacmod::gstore](#)
- `real(rprec), dimension(:, :), allocatable` [vacmod::green](#)
- `real(rprec), dimension(:, :), allocatable` [vacmod::greenp](#)
- `real(rprec), dimension(:), allocatable` [vacmod::r0p](#)
- `real(rprec), dimension(:), allocatable` [vacmod::r1p](#)
- `real(rprec), dimension(:), allocatable` [vacmod::r0m](#)
- `real(rprec), dimension(:), allocatable` [vacmod::r1m](#)
- `real(rprec), dimension(:), allocatable` [vacmod::sqrtc](#)

- real(rprec), dimension(:), allocatable [vacmod::sqrra](#)
- real(rprec), dimension(:), allocatable [vacmod::tlp2](#)
- real(rprec), dimension(:), allocatable [vacmod::tlp1](#)
- real(rprec), dimension(:), allocatable [vacmod::tlp](#)
- real(rprec), dimension(:), allocatable [vacmod::tlm2](#)
- real(rprec), dimension(:), allocatable [vacmod::tlm1](#)
- real(rprec), dimension(:), allocatable [vacmod::tlm](#)
- real(rprec), dimension(:), allocatable [vacmod::adp](#)
- real(rprec), dimension(:), allocatable [vacmod::adm](#)
- real(rprec), dimension(:), allocatable [vacmod::cma](#)
- real(rprec), dimension(:), allocatable [vacmod::ra1p](#)
- real(rprec), dimension(:), allocatable [vacmod::ra1m](#)
- real(rprec), dimension(:), allocatable [vacmod::slm](#)
- real(rprec), dimension(:), allocatable [vacmod::slp](#)
- real(rprec), dimension(:), allocatable [vacmod::tlpm](#)
- real(rprec), dimension(:), allocatable [vacmod::slpm](#)
- real(rprec), dimension(:), allocatable [vacmod::delt1u](#)
- real(rprec), dimension(:), allocatable [vacmod::azp1u](#)
- real(rprec), dimension(:), allocatable [vacmod::azm1u](#)
- real(rprec), dimension(:), allocatable [vacmod::cma11u](#)
- real(rprec), dimension(:), allocatable [vacmod::sqad1u](#)
- real(rprec), dimension(:), allocatable [vacmod::sqad2u](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::all_tlp](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::all_tlm](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::all_slp](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::all_slm](#)
- real(rprec), dimension(:), allocatable [vacmod::gsave](#)
- real(rprec), dimension(:), allocatable [vacmod::ga1](#)
- real(rprec), dimension(:), allocatable [vacmod::ga2](#)
- real(rprec), dimension(:), allocatable [vacmod::dsave](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::g1](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::g2](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::bcos](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::bsin](#)
- real(rprec), dimension(:,,:), allocatable [vacmod::source](#)
- real(rprec), dimension(:,::,:), allocatable [vacmod::actemp](#)
- real(rprec), dimension(:,::,:), allocatable [vacmod::astemp](#)

7.71 src/NESTOR/data/vacmod0.f90 File Reference

Modules

- module [vacmod0](#)

Functions/Subroutines

- subroutine [vacmod0::set_nestor_sizes](#) (nfp, ntor, mpol, nzeta, ntheta, lasym)

Variables

- integer `vacmod0::mf`
- integer `vacmod0::nf`
- integer `vacmod0::nu`
- integer `vacmod0::nv`
- integer `vacmod0::mf1`
- integer `vacmod0::nf1`
- integer `vacmod0::mnpd`
- integer `vacmod0::mnpd2`
- integer `vacmod0::nuv`
- integer `vacmod0::nu2`
- integer `vacmod0::nu3`
- integer `vacmod0::nuv2`
- integer `vacmod0::nfper`
- integer `vacmod0::nvper`
- integer `vacmod0::nuv_tan`
- integer `vacmod0::nvp`
- integer `vacmod0::ndim`

7.72 src/NESTOR/fouri.f90 File Reference

Functions/Subroutines

- subroutine `fouri` (`grpmn`, `gsource`, `amatrix`, `amatsq`, `bvec`, `wint`, `lasym`)

7.72.1 Function/Subroutine Documentation

7.72.1.1 `fouri()`

```

subroutine fouri (
    real(rprec), dimension(mnpd,nv,nu3,ndim), intent(in) grpmn,
    real(rprec), dimension(nuv), intent(in) gsource,
    real(rprec), dimension(mnpd,mnpd,ndim**2), intent(out) amatrix,
    real(rprec), dimension(mnpd2,mnpd2), intent(out) amatsq,
    real(rprec), dimension(0:mf,-nf:nf,ndim), intent(inout) bvec,
    real(rprec), dimension(nuv2), intent(in) wint,
    logical, intent(in) lasym )

```

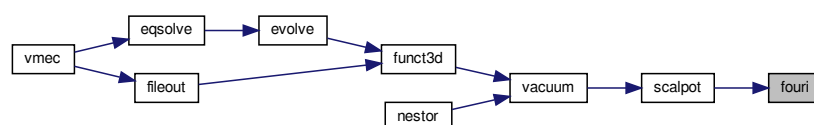
interior (`int_ext=-1`), exterior (`int_ext=+1`) neumann problem

Definition at line 2 of file `fouri.f90`.

References `vacmod::actemp`, `vacmod::amatrix`, `vacmod::astemp`, `vacmod::bcos`, `vacmod::bsin`, `vacmod::grpmn`, `vacmod::onp`, `vacmod::p5`, `vacmod::pi2`, `vacmod::pi3`, and `vacmod::source`.

Referenced by `scalpot()`.

Here is the caller graph for this function:



7.73 src/NESTOR/fourp.f90 File Reference

Functions/Subroutines

- subroutine [fourp](#) (grpmn, grp)

7.73.1 Function/Subroutine Documentation

7.73.1.1 fourp()

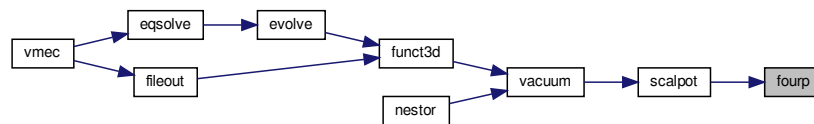
```
subroutine fourp (
    real(rprec), dimension(0:mf,-nf:nf,nuv2,ndim), intent(inout) grpmn,
    real(rprec), dimension(nuv,nuv2), intent(in) grp )
```

Definition at line 2 of file fourp.f90.

References `vacmod::g1`, `vacmod::g2`, `vacmod::grpmn`, `vacmod::onp`, and `vacmod::p5`.

Referenced by `scalpot()`.

Here is the caller graph for this function:



7.74 src/NESTOR/greenf.f90 File Reference

Functions/Subroutines

- subroutine [greenf](#) (delgr, delgrp, ip)

7.74.1 Function/Subroutine Documentation

7.74.1.1 greenf()

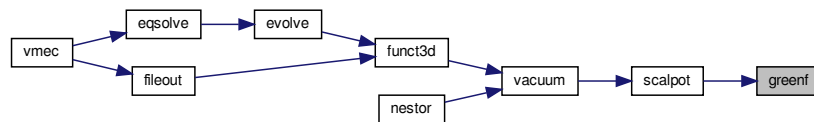
```
subroutine greenf (
    real(rprec), dimension(nuv), intent(out) delgr,
    real(rprec), dimension(nuv), intent(out) delgrp,
    integer, intent(in) ip )
```

Definition at line 2 of file greenf.f90.

References vacmod::auu, vacmod::auv, vacmod::avv, vacmod::drv, vacmod::dsave, vacmod::ga1, vacmod::ga2, vacmod::gsave, vacmod::guu_b, vacmod::guv_b, vacmod::gvv_b, vacmod::r1b, vacmod::rcosuv, vacmod::rsinuv, vacmod::rzb2, vacmod::snr, vacmod::snv, vacmod::snz, and vacmod::z1b.

Referenced by scalpot().

Here is the caller graph for this function:



7.75 src/NESTOR/nestor_main.f90 File Reference

Main program of stand-alone version of NESTOR.

Functions/Subroutines

- program [nestor](#)
Main program of stand-alone version of NESTOR.

7.75.1 Detailed Description

Main program of stand-alone version of NESTOR.

7.75.2 Function/Subroutine Documentation

7.75.2.1 nestor()

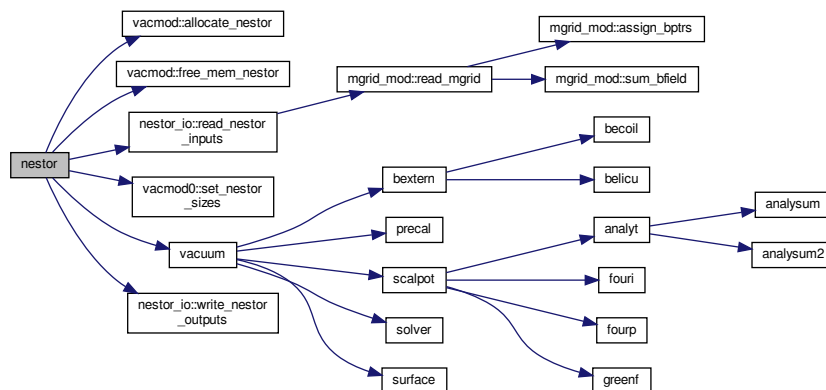
```
program nestor
```

Main program of stand-alone version of NESTOR.

Definition at line 5 of file nestor_main.f90.

References `vacmod::allocate_nestor()`, `vacmod::amatsav`, `nestor_io::amatsav_nestor`, `vacmod::bsubvvac`, `nestor_io::bsubvvac_nestor`, `vacmod::bvecsav`, `nestor_io::bvecsav_nestor`, `nestor_io::ctor`, `stel_kinds::dp`, `vacmod::free_mem_nestor()`, `nestor_io::ier_flag`, `nestor_io::input_extension`, `nestor_io::ivac`, `nestor_io::ivacskip`, `nestor_io::lasym`, `nestor_io::mnmax`, `nestor_io::mpol`, `nestor_io::nfp`, `nestor_io::ntheta`, `nestor_io::ntor`, `nestor_io::nzeta`, `nestor_io::raxis`, `nestor_io::rbtor`, `nestor_io::read_nestor_inputs()`, `nestor_io::rmnc`, `nestor_io::rmns`, `vacmod0::set_nestor_sizes()`, `nestor_io::signgs`, `vacuum()`, `nestor_io::vacuum_calls`, `nestor_io::wint`, `nestor_io::write_nestor_outputs()`, `nestor_io::xm`, `nestor_io::xn`, `nestor_io::zaxis`, `nestor_io::zmnc`, and `nestor_io::zmns`.

Here is the call graph for this function:



7.76 src/NESTOR/precalf90 File Reference

Functions/Subroutines

- subroutine [precalf](#)

7.76.1 Function/Subroutine Documentation

7.76.1.1 precal()

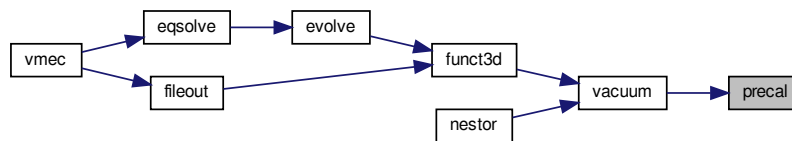
```
subroutine precal
```

Definition at line 2 of file precal.f90.

References `vacmod::alp`, `vacmod::alu`, `vacmod::alv`, `vacmod::alvp`, `vacmod::onp`, `vacmod::onp2`, `vacmod::p5`, `vacmod::pi2`, `vacmod::pi3`, `vacmod::pi4`, and `vacmod::precal_done`.

Referenced by `vacuum()`.

Here is the caller graph for this function:



7.77 src/NESTOR/scalpot.f90 File Reference

Functions/Subroutines

- subroutine `scalpot` (`bvec`, `amatrix`, `wint`, `ivacskip`, `lasym`, `m_map`, `n_map`)

7.77.1 Function/Subroutine Documentation

7.77.1.1 scalpot()

```

subroutine scalpot (
    real(rprec), dimension(mnpg2), intent(out) bvec,
    real(rprec), dimension(mnpg2*mnpg2), intent(out) amatrix,
    real(rprec), dimension(nuv2), intent(in) wint,
    integer, intent(in) ivacskip,
    logical, intent(in) lasym,
    real(rprec), dimension(mnpg2), intent(out) m_map,
    real(rprec), dimension(mnpg2), intent(out) n_map )

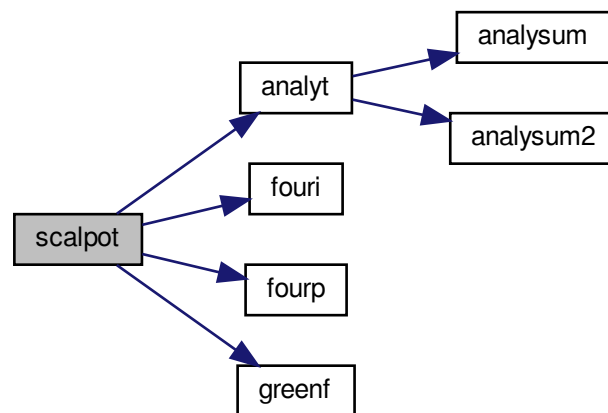
```

Definition at line 2 of file scalpot.f90.

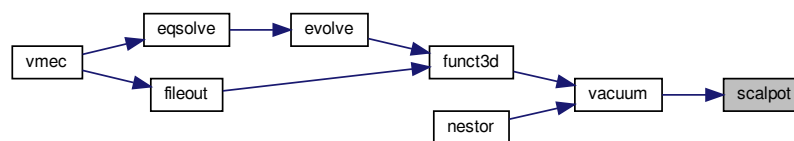
References `vacmod::amatrix`, `vacmod::amatsav`, `analyt()`, `vacmod::bexni`, `vacmod::bvecsav`, `four()`, `fourp()`, `vacmod::green`, `greenf()`, `vacmod::greenp`, `vacmod::grpmn`, `vacmod::grpmn_m_map_wrt`, `vacmod::grpmn_n_map_wrt`, and `vacmod::gstore`.

Referenced by `vacuum()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.78 src/NESTOR/surface.f90 File Reference

Functions/Subroutines

- subroutine [surface](#) (rc, rs, zs, zc, xm, xn, mnmax, lasym, signsgs)

7.78.1 Function/Subroutine Documentation

7.78.1.1 surface()

```

subroutine surface (
    real(rprec), dimension(mnmax), intent(in) rc,
    real(rprec), dimension(mnmax), intent(in) rs,
    real(rprec), dimension(mnmax), intent(in) zs,
    real(rprec), dimension(mnmax), intent(in) zc,
    real(rprec), dimension(mnmax), intent(in) xm,
    real(rprec), dimension(mnmax), intent(in) xn,
    integer, intent(in) mnmax,
    logical, intent(in) lasym,
    real(rprec), intent(in) signgs )

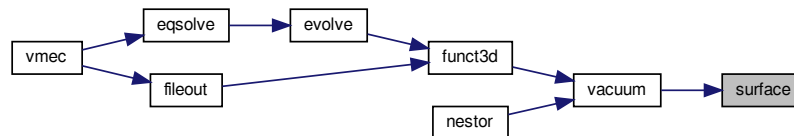
```

Definition at line 2 of file surface.f90.

References vacmod::auu, vacmod::auv, vacmod::avv, vacmod::drv, vacmod::guu_b, vacmod::guv_b, vacmod::gvv_b, vacmod::onp, vacmod::onp2, vacmod::p5, vacmod::r1b, vacmod::rcosuv, vacmod::rsinuv, vacmod::rub, vacmod::ruu, vacmod::ruv, vacmod::rvb, vacmod::rvv, vacmod::rzb2, vacmod::snr, vacmod::snv, vacmod::snz, vacmod::z1b, vacmod::zub, vacmod::zuu, vacmod::zuv, vacmod::zvb, and vacmod::zvv.

Referenced by vacuum().

Here is the caller graph for this function:



7.79 src/NESTOR/vacuum.f90 File Reference

Functions/Subroutines

- subroutine [vacuum](#) (rmnc, rmns, zmns, zmnc, xm, xn, plascur, rbtor, wint, ivac_skip, ivac, mnmax, ier_flag, lasym, signgs, raxis, zaxis)

7.79.1 Function/Subroutine Documentation

7.79.1.1 vacuum()

```

subroutine vacuum (
    real(rprec), dimension(mnmax), intent(in)  rmnc,
    real(rprec), dimension(mnmax), intent(in)  rmns,
    real(rprec), dimension(mnmax), intent(in)  zmns,
    real(rprec), dimension(mnmax), intent(in)  zmnc,
    real(rprec), dimension(mnmax), intent(in)  xm,
    real(rprec), dimension(mnmax), intent(in)  xn,
    real(rprec), intent(in)  plascur,
    real(rprec), intent(in)  rbtor,
    real(rprec), dimension(nuv2), intent(in)  wint,
    integer, intent(in)  ivac_skip,
    integer, intent(inout)  ivac,
    integer, intent(in)  mnmax,
    integer, intent(inout)  ier_flag,
    logical, intent(in)  lasym,
    real(rprec), intent(in)  signs,
    real(rprec), dimension(nv), intent(in)  raxis,
    real(rprec), dimension(nv), intent(in)  zaxis )

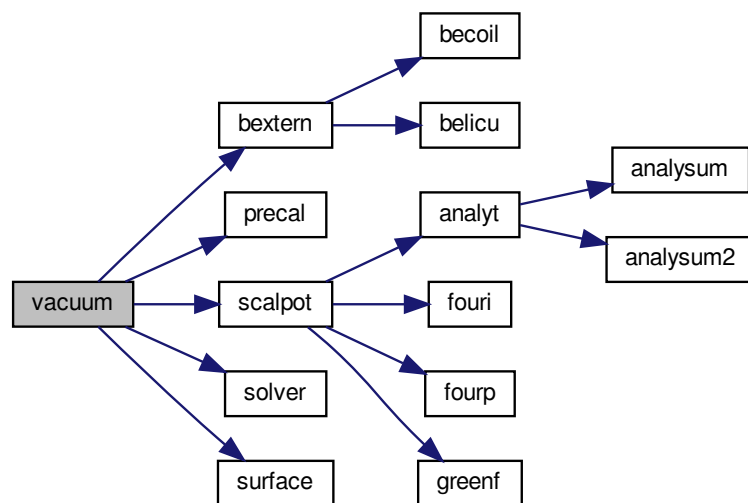
```

Definition at line 2 of file vacuum.f90.

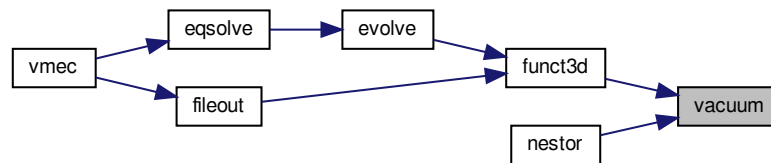
References vacmod::amatrix, bextern(), vacmod::bexu, vacmod::bexv, vacmod::bphiv, vacmod::brv, vacmod::bsqvac, vacmod::bsubu, vacmod::bsubv, vacmod::bsubvvac, vacmod::bzv, vacmod::guu_b, vacmod::guv_b, vacmod::gvv_b, vacmod::m_map_wrt, vacmod::n_map_wrt, vmec_params::norm_term_flag, vacmod::p5, vmec_params::phiedge_error_flag, vacmod::pi2, vacmod::potu, vacmod::potv, vacmod::potvac, precal(), vacmod::precal_done, vacmod::r1b, vacmod::raxis_nestor, vacmod::rub, vacmod::rvb, scalpot(), solver(), surface(), vacmod::zaxis_nestor, vacmod::zub, and vacmod::zvb.

Referenced by funct3d(), and nestor().

Here is the call graph for this function:



Here is the caller graph for this function:



7.80 src/open_output_files.f90 File Reference

Open output files.

Functions/Subroutines

- subroutine [open_output_files](#) (extension, lfirst)
Open output files.

7.80.1 Detailed Description

Open output files.

7.80.2 Function/Subroutine Documentation

7.80.2.1 open_output_files()

```

subroutine open_output_files (
    character(len=*) extension,
    logical lfirst )

```

Open output files.

Parameters

<i>extension</i>	input file "extension": part after 'input.'.
<i>lfirst</i>	flag to indicate if this is the first call to this routine or not

Definition at line 8 of file open_output_files.f90.

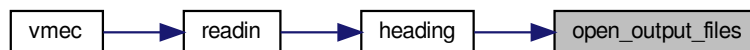
References `vparams::nthreed`, `vparams::nthreed0`, and `safe_open_mod::safe_open()`.

Referenced by heading().

Here is the call graph for this function:



Here is the caller graph for this function:



7.81 src/parse_extension.f File Reference

Parse the first command-line argument into a filename.

Functions/Subroutines

- subroutine `parse_extension` (`file_to_parse`, `file_or_extension`, `lnc`)
Parse the first command-line argument into a filename.

7.81.1 Detailed Description

Parse the first command-line argument into a filename.

7.81.2 Function/Subroutine Documentation

7.81.2.1 parse_extension()

```

subroutine parse_extension (
    character(len=*), intent(inout) file_to_parse,
    character(len=*), intent(in) file_or_extension,
    logical, intent(out) lnc )
  
```

Parse the first command-line argument into a filename.

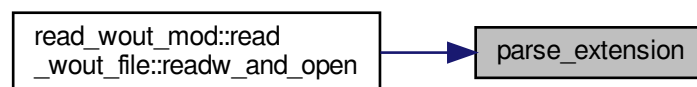
Parameters

<i>file_to_parse</i>	actual filename to read the input for VMEC from
<i>file_or_extension</i>	first command-line parameter given to VMEC
<i>Inc</i>	flag to indicate that a netCDF file is given

Definition at line 9 of file parse_extension.f.

Referenced by read_wout_mod::read_wout_file::readw_and_open().

Here is the caller graph for this function:



7.82 src/precondn.f90 File Reference

Compute preconditioning matrix elements for R , Z force.

Functions/Subroutines

- subroutine [precondn](#) (lu1, bsq, gsqrt, r12, xs, xu12, xue, xuo, xodd, axm, axd, bxm, bxd, cx, eqfactor, trigmult)
Compute preconditioning matrix elements for R , Z force.

7.82.1 Detailed Description

Compute preconditioning matrix elements for R , Z force.

7.82.2 Function/Subroutine Documentation

7.82.2.1 precondn()

```

subroutine precondn (
    real(rprec), dimension(nrzt), intent(in) lu1,
    real(rprec), dimension(nrzt), intent(in) bsq,
    real(rprec), dimension(nrzt), intent(in) gsqrt,
    real(rprec), dimension(nrzt), intent(in) r12,
    real(rprec), dimension(nrzt), intent(in) xs,
    real(rprec), dimension(nrzt), intent(in) xu12,
    real(rprec), dimension(nrzt), intent(in) xue,
    real(rprec), dimension(nrzt), intent(in) xuo,
    real(rprec), dimension(nrzt), intent(in) xodd,
    real(rprec), dimension(ns+1,2), intent(out) axm,
    real(rprec), dimension(ns+1,2), intent(out) axd,
    real(rprec), dimension(ns+1,2), intent(out) bxm,
    real(rprec), dimension(ns+1,2), intent(out) bxd,
    real(rprec), dimension(ns+1), intent(out) cx,
    real(rprec), dimension(ns), intent(out) eqfactor,
    real(rprec), dimension(nznt), intent(in) trigmult )

```

Compute preconditioning matrix elements for R , Z force.

Parameters

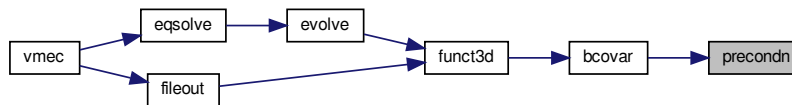
<i>lu1</i>	
<i>bsq</i>	
<i>gsqrt</i>	
<i>r12</i>	
<i>xs</i>	
<i>xu12</i>	
<i>xue</i>	
<i>xuo</i>	
<i>xodd</i>	
<i>axm</i>	
<i>axd</i>	
<i>bxm</i>	
<i>bxd</i>	
<i>cx</i>	
<i>eqfactor</i>	
<i>trigmult</i>	

Definition at line 22 of file precondn.f90.

References vmec_main::hs, vmec_main::ohs, vmec_main::r0scale, realspace::shalf, vmec_params::signgs, vmec_main::sm, vmec_main::sp, vmec_main::vp, and realspace::wint.

Referenced by bcovar().

Here is the caller graph for this function:



7.83 src/printout.f90 File Reference

Print iteration progress to screen and `threed1` output file.

Functions/Subroutines

- subroutine `printout` (`i0`, `delt0`, `w0`)
Print iteration progress to screen and `threed1` output file.

7.83.1 Detailed Description

Print iteration progress to screen and `threed1` output file.

7.83.2 Function/Subroutine Documentation

7.83.2.1 `printout()`

```

subroutine printout (
    integer i0,
    real(rprec) delt0,
    real(rprec) w0 )

```

Print iteration progress to screen and `threed1` output file.

Parameters

<i>i0</i>	current iteration number
<i>delt0</i>	current time step
<i>w0</i>	current MHD energy

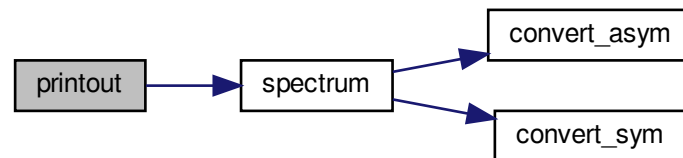
Definition at line 9 of file `printout.f90`.

References `vmec_main::bsqsav`, `vmec_main::dbsq`, `vmec_main::delbsq`, `vmec_main::fedge`, `vmec_main::fsql`,

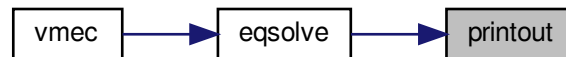
vmec_main::fsql1, vmec_main::fsqr, vmec_main::fsqr1, vmec_main::fsqz, vmec_main::fsqz1, xstuff::gc, vmec_main::irzloff, vmec_main::iter2, vmec_main::ivac, vmec_main::r00, spectrum(), vmec_main::specw, vmec_main::vp, vmec_main::wb, realspace::wint, vmec_main::wp, xstuff::xstore, and vmec_main::z00.

Referenced by eqsolve().

Here is the call graph for this function:



Here is the caller graph for this function:



7.84 src/profil1d.f90 File Reference

Compute phip and iota profiles on full grid.

Functions/Subroutines

- subroutine [profil1d](#) (xc, xcdot, lreset)
Compute phip and iota profiles on full grid.

7.84.1 Detailed Description

Compute phip and iota profiles on full grid.

7.84.2 Function/Subroutine Documentation

7.84.2.1 profil1d()

```
subroutine profil1d (
    real(rprec), dimension(negs), intent(out) xc,
    real(rprec), dimension(negs), intent(out) xcdot,
    logical, intent(in) lreset )
```

Compute phip and iota profiles on full grid.

Parameters

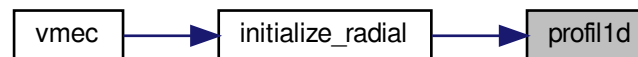
<i>xc</i>	state vector of VMEC, i.e., all Fourier coefficients of R , Z and λ
<i>xcdot</i>	velocity vector in Fourier space
<i>lreset</i>	xc will be zeroes if this is true

Definition at line 9 of file profil1d.f90.

References vmec_main::bdamp, vmec_main::chipf, vmec_main::chips, vmec_main::currv, vmec_main::hs, vmec_main::icurv, vmec_main::iotaf, vmec_main::iotas, vmec_params::lamscale, vmec_main::lflip, vmec_main::mass, vmec_params::pdamp, vmec_main::phipf, vmec_main::phips, vmec_main::pres, vmec_main::r00, vmec_params::rcc, vmec_main::rmn_bdy, realspace::shalf, vmec_params::signgs, vmec_main::sm, vmec_main::sp, and realspace::sqrts.

Referenced by initialize_radial().

Here is the caller graph for this function:



7.85 src/profil3d.f90 File Reference

Compute three-dimensional profiles of flux-surface geometry etc.

Functions/Subroutines

- subroutine [profil3d](#) (rmn, zmn, lreset)
Compute three-dimensional profiles of flux-surface geometry etc.

7.85.1 Detailed Description

Compute three-dimensional profiles of flux-surface geometry etc.

7.85.2 Function/Subroutine Documentation

7.85.2.1 profil3d()

```
subroutine profil3d (
    real(rprec), dimension(ns,0:ntor,0:mpoll,ntmax), intent(inout) rmn,
    real(rprec), dimension(ns,0:ntor,0:mpoll,ntmax), intent(inout) zmn,
    logical, intent(in) lreset )
```

Compute three-dimensional profiles of flux-surface geometry etc.

Parameters

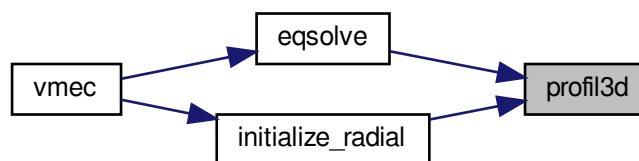
<i>rmn</i>	Fourier coefficients of R
<i>zmn</i>	Fourier coefficients of R
<i>lreset</i>	flag to indicate the geometry of the LCFS (and axis ?) should be used to interpolate into the plasma volume

Definition at line 9 of file profil3d.f90.

References `realspace::chip`, `vmec_main::chips`, `vmec_main::faclam`, `vmec_main::ireflect`, `vmec_main::irzloff`, `vmec_params::mscale`, `vmec_params::nscale`, `vmec_params::ntmax`, `realspace::phip`, `vmec_main::phips`, `vmec_params::rcc`, `vmec_params::rcs`, `vmec_main::rmn_bdy`, `xstuff::scalxc`, `realspace::sqrts`, `vmec_params::uminus`, `realspace::wint`, `vmec_params::zcc`, `vmec_params::zcs`, and `vmec_main::zmn_bdy`.

Referenced by `eqsolve()`, and `initialize_radial()`.

Here is the caller graph for this function:



7.86 src/profile_functions.f File Reference

Parameterizations of profiles for rotational transform, toroidal current and pressure/mass.

Functions/Subroutines

- real(rprec) function `pcurr` (xx)
- real(rprec) function `piota` (x)
- real(rprec) function `pmass` (xx)

7.86.1 Detailed Description

Parameterizations of profiles for rotational transform, toroidal current and pressure/mass.

7.86.2 Function/Subroutine Documentation

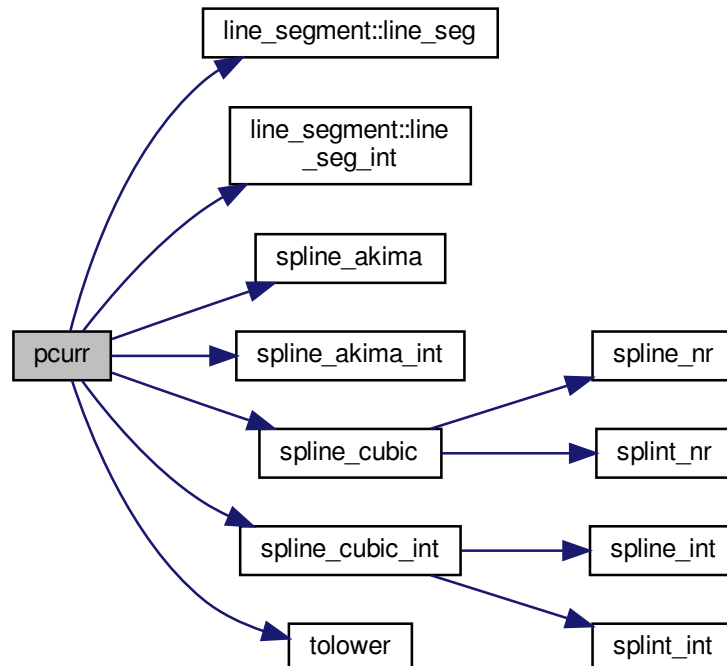
7.86.2.1 pcurr()

```
real(rprec) function pcurr (
    real(rprec) xx )
```

Definition at line 11 of file profile_functions.f.

References vmec_input::ac, vmec_input::ac_aux_f, vmec_input::ac_aux_s, vmec_input::bloat, line_segment::line_seg(), line_segment::line_seg_int(), stel_constants::one, vmec_input::pcurr_type, stel_constants::pi, spline_akima(), spline_akima_int(), spline_cubic(), spline_cubic_int(), tolower(), and stel_constants::zero.

Here is the call graph for this function:



7.86.2.2 piota()

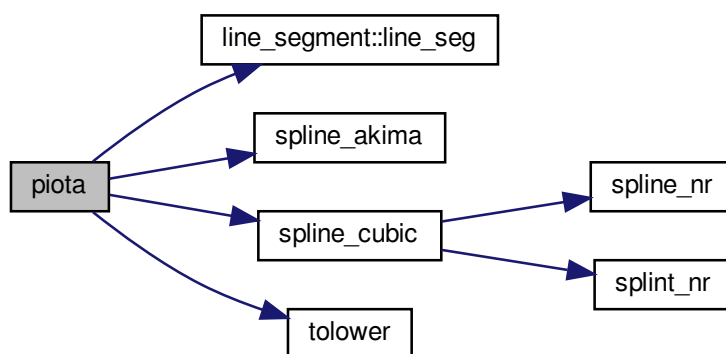
```
real(rprec) function piota (  
    real(rprec) x )
```

Definition at line 323 of file profile_functions.f.

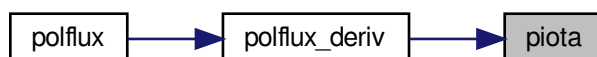
References vmec_input::ai, vmec_input::ai_aux_f, vmec_input::ai_aux_s, line_segment::line_seg(), stel_↵ constants::one, stel_constants::pi, vmec_input::piota_type, spline_akima(), spline_cubic(), tolower(), and stel_↵ _constants::zero.

Referenced by polflux_deriv().

Here is the call graph for this function:



Here is the caller graph for this function:



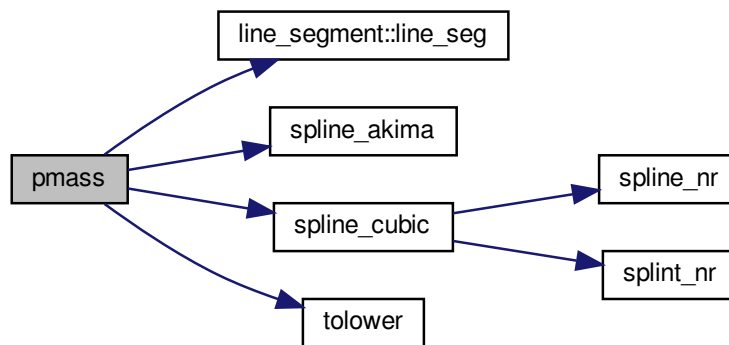
7.86.2.3 pmass()

```
real(rprec) function pmass (
    real(rprec) xx )
```

Definition at line 464 of file profile_functions.f.

References vmec_input::am, vmec_input::am_aux_f, vmec_input::am_aux_s, vmec_input::bloat, line_segment::line_seg(), stel_constants::one, vmec_input::pmass_type, vmec_input::pres_scale, spline_akima(), spline_cubic(), tolower(), and stel_constants::zero.

Here is the call graph for this function:



7.87 src/read_indata.f90 File Reference

Read the INDATA namelist from a given input file.

Functions/Subroutines

- subroutine [read_indata](#) (in_file, iunit, ier_flag)
Read the INDATA namelist from a given input file.

7.87.1 Detailed Description

Read the INDATA namelist from a given input file.

7.87.2 Function/Subroutine Documentation

7.87.2.1 read_indata()

```
subroutine read_indata (
    character(len=*) in_file,
    integer iunit,
    integer ier_flag )
```

Read the INDATA namelist from a given input file.

Parameters

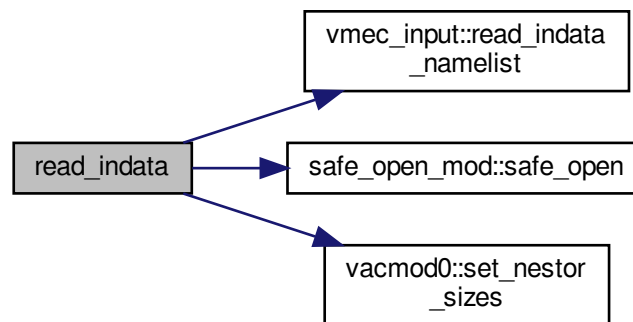
<i>in_file</i>	input file to read from
<i>iunit</i>	unit number to use for input file
<i>ier_flag</i>	error flag

Definition at line 9 of file read_indata.f90.

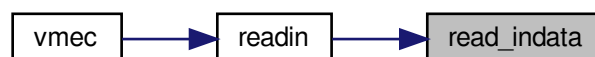
References vmec_input::bloat, vmec_params::input_error_flag, vmec_main::lthreed, vmec_input::ncurr, vmec_params::ntmax, vmec_params::rcc, vmec_params::rcs, vmec_input::read_indata_namelist(), vmec_params::rsc, vmec_params::rss, safe_open_mod::safe_open(), vacmod0::set_nestor_sizes(), vmec_params::zcc, vmec_params::zcs, vmec_params::zsc, and vmec_params::zss.

Referenced by readin().

Here is the call graph for this function:



Here is the caller graph for this function:



7.88 src/read_wout_mod.f File Reference

Reading of `wout` VMEC output file.

Data Types

- interface [read_wout_mod::read_wout_file](#)

Modules

- module [read_wout_mod](#)
Reading of wout VMEC output file.

Functions/Subroutines

- subroutine [read_wout_mod::readw_and_open](#) (file_or_extension, ierr, iopen)
- subroutine [read_wout_mod::compute_currents](#) (ierror)
- subroutine [read_wout_mod::read_wout_deallocate](#)
- subroutine [read_wout_mod::tosuvspace](#) (s_in, u_in, v_in, gsqr, bsupu, bsupv, jsupu, jsupv, lam)
- subroutine [read_wout_mod::loadrzl](#)

Variables

- character(len= *), parameter [read_wout_mod::vn_version](#) = 'version_'
- character(len= *), parameter [read_wout_mod::vn_extension](#) = 'input_extension'
- character(len= *), parameter [read_wout_mod::vn_mgrid](#) = 'mgrid_file'
- character(len= *), parameter [read_wout_mod::vn_magen](#) = 'wb'
- character(len= *), parameter [read_wout_mod::vn_therm](#) = 'wp'
- character(len= *), parameter [read_wout_mod::vn_gam](#) = 'gamma'
- character(len= *), parameter [read_wout_mod::vn_maxr](#) = 'rmax_surf'
- character(len= *), parameter [read_wout_mod::vn_minr](#) = 'rmin_surf'
- character(len= *), parameter [read_wout_mod::vn_maxz](#) = 'zmax_surf'
- character(len= *), parameter [read_wout_mod::vn_fp](#) = 'nfp'
- character(len= *), parameter [read_wout_mod::vn_radnod](#) = 'ns'
- character(len= *), parameter [read_wout_mod::vn_polmod](#) = 'mpol'
- character(len= *), parameter [read_wout_mod::vn_tormod](#) = 'ntor'
- character(len= *), parameter [read_wout_mod::vn_maxmod](#) = 'mnmax'
- character(len= *), parameter [read_wout_mod::vn_maxit](#) = 'niter'
- character(len= *), parameter [read_wout_mod::vn_actit](#) = 'itfsq'
- character(len= *), parameter [read_wout_mod::vn_asym](#) = 'lasym'
- character(len= *), parameter [read_wout_mod::vn_free](#) = 'lfreeb'
- character(len= *), parameter [read_wout_mod::vn_error](#) = 'ier_flag'
- character(len= *), parameter [read_wout_mod::vn_aspect](#) = 'aspect'
- character(len= *), parameter [read_wout_mod::vn_maxmod_nyq](#) = 'mnmax_nyq'
- character(len= *), parameter [read_wout_mod::vn_beta](#) = 'betatotal'
- character(len= *), parameter [read_wout_mod::vn_pbeta](#) = 'betapol'
- character(len= *), parameter [read_wout_mod::vn_tbeta](#) = 'betator'
- character(len= *), parameter [read_wout_mod::vn_abeta](#) = 'betaxis'
- character(len= *), parameter [read_wout_mod::vn_b0](#) = 'b0'
- character(len= *), parameter [read_wout_mod::vn_rbt0](#) = 'rbtor0'
- character(len= *), parameter [read_wout_mod::vn_rbt1](#) = 'rbtor'
- character(len= *), parameter [read_wout_mod::vn_sgs](#) = 'signgs'
- character(len= *), parameter [read_wout_mod::vn_lar](#) = 'lonLarmor'
- character(len= *), parameter [read_wout_mod::vn_modb](#) = 'volavgB'
- character(len= *), parameter [read_wout_mod::vn_ctor](#) = 'ctor'
- character(len= *), parameter [read_wout_mod::vn_amin](#) = 'Aminor_p'

- character(len= *), parameter `read_wout_mod::vn_rmaj` = 'Rmajor_p'
- character(len= *), parameter `read_wout_mod::vn_vol` = 'volume_p'
- character(len= *), parameter `read_wout_mod::vn_am` = 'am'
- character(len= *), parameter `read_wout_mod::vn_ai` = 'ai'
- character(len= *), parameter `read_wout_mod::vn_ac` = 'ac'
- character(len= *), parameter `read_wout_mod::vn_ah` = 'hot particle fraction'
- character(len= *), parameter `read_wout_mod::vn_atuname` = 'T-perp/T-par'
- character(len= *), parameter `read_wout_mod::vn_pmass_type` = 'pmass_type'
- character(len= *), parameter `read_wout_mod::vn_piota_type` = 'piota_type'
- character(len= *), parameter `read_wout_mod::vn_pcurr_type` = 'pcurr_type'
- character(len= *), parameter `read_wout_mod::vn_am_aux_s` = 'am_aux_s'
- character(len= *), parameter `read_wout_mod::vn_am_aux_f` = 'am_aux_f'
- character(len= *), parameter `read_wout_mod::vn_ai_aux_s` = 'ai_aux_s'
- character(len= *), parameter `read_wout_mod::vn_ai_aux_f` = 'ai_aux_f'
- character(len= *), parameter `read_wout_mod::vn_ac_aux_s` = 'ac_aux_s'
- character(len= *), parameter `read_wout_mod::vn_ac_aux_f` = 'ac_aux_f'
- character(len= *), parameter `read_wout_mod::vn_mse` = 'imse'
- character(len= *), parameter `read_wout_mod::vn_thom` = 'itse'
- character(len= *), parameter `read_wout_mod::vn_pmod` = 'xm'
- character(len= *), parameter `read_wout_mod::vn_tmod` = 'xn'
- character(len= *), parameter `read_wout_mod::vn_pmod_nyq` = 'xm_nyq'
- character(len= *), parameter `read_wout_mod::vn_tmod_nyq` = 'xn_nyq'
- character(len= *), parameter `read_wout_mod::vn_racc` = 'raxis_cc'
- character(len= *), parameter `read_wout_mod::vn_zacs` = 'zaxis_cs'
- character(len= *), parameter `read_wout_mod::vn_racs` = 'raxis_cs'
- character(len= *), parameter `read_wout_mod::vn_zacc` = 'zaxis_cc'
- character(len= *), parameter `read_wout_mod::vn_iotaf` = 'iotaf'
- character(len= *), parameter `read_wout_mod::vn_qfact` = 'q-factor'
- character(len= *), parameter `read_wout_mod::vn_chi` = 'chi'
- character(len= *), parameter `read_wout_mod::vn_chipf` = 'chipf'
- character(len= *), parameter `read_wout_mod::vn_presf` = 'presf'
- character(len= *), parameter `read_wout_mod::vn_phi` = 'phi'
- character(len= *), parameter `read_wout_mod::vn_phipf` = 'phipf'
- character(len= *), parameter `read_wout_mod::vn_jcuru` = 'jcuru'
- character(len= *), parameter `read_wout_mod::vn_jcurv` = 'jcurv'
- character(len= *), parameter `read_wout_mod::vn_iotah` = 'iotas'
- character(len= *), parameter `read_wout_mod::vn_mass` = 'mass'
- character(len= *), parameter `read_wout_mod::vn_presh` = 'pres'
- character(len= *), parameter `read_wout_mod::vn_betah` = 'beta_vol'
- character(len= *), parameter `read_wout_mod::vn_buco` = 'buco'
- character(len= *), parameter `read_wout_mod::vn_bvco` = 'bvco'
- character(len= *), parameter `read_wout_mod::vn_vp` = 'vp'
- character(len= *), parameter `read_wout_mod::vn_specw` = 'specw'
- character(len= *), parameter `read_wout_mod::vn_hiph` = 'phips'
- character(len= *), parameter `read_wout_mod::vn_jdotb` = 'jdotb'
- character(len= *), parameter `read_wout_mod::vn_overr` = 'over_r'
- character(len= *), parameter `read_wout_mod::vn_bgrv` = 'bdotgradv'
- character(len= *), parameter `read_wout_mod::vn_merc` = 'DMerc'
- character(len= *), parameter `read_wout_mod::vn_mshear` = 'DShear'
- character(len= *), parameter `read_wout_mod::vn_mwell` = 'DWell'
- character(len= *), parameter `read_wout_mod::vn_mcurr` = 'DCurr'
- character(len= *), parameter `read_wout_mod::vn_mgeo` = 'DGeod'
- character(len= *), parameter `read_wout_mod::vn_equip` = 'equip'
- character(len= *), parameter `read_wout_mod::vn_fsqt` = 'fsqt'
- character(len= *), parameter `read_wout_mod::vn_wdot` = 'wdot'

- character(len= *), parameter read_wout_mod::vn_ftolv = 'ftolv'
- character(len= *), parameter read_wout_mod::vn_fsqli = 'fsqli'
- character(len= *), parameter read_wout_mod::vn_fsqr = 'fsqr'
- character(len= *), parameter read_wout_mod::vn_fsqz = 'fsqz'
- character(len= *), parameter read_wout_mod::vn_extcur = 'extcur'
- character(len= *), parameter read_wout_mod::vn_curlab = 'curlabel'
- character(len= *), parameter read_wout_mod::vn_rmnc = 'rmnc'
- character(len= *), parameter read_wout_mod::vn_zmns = 'zmns'
- character(len= *), parameter read_wout_mod::vn_lmns = 'lmns'
- character(len= *), parameter read_wout_mod::vn_gmnc = 'gmnc'
- character(len= *), parameter read_wout_mod::vn_bmnc = 'bmnc'
- character(len= *), parameter read_wout_mod::vn_bsubumnc = 'bsubumnc'
- character(len= *), parameter read_wout_mod::vn_bsubvmnc = 'bsubvmnc'
- character(len= *), parameter read_wout_mod::vn_bsubsmns = 'bsubsmns'
- character(len= *), parameter read_wout_mod::vn_bsupumnc = 'bsupumnc'
- character(len= *), parameter read_wout_mod::vn_bsupvmnc = 'bsupvmnc'
- character(len= *), parameter read_wout_mod::vn_rmns = 'rmns'
- character(len= *), parameter read_wout_mod::vn_zmnc = 'zmnc'
- character(len= *), parameter read_wout_mod::vn_lmnc = 'lmnc'
- character(len= *), parameter read_wout_mod::vn_gmns = 'gmns'
- character(len= *), parameter read_wout_mod::vn_bmns = 'bmns'
- character(len= *), parameter read_wout_mod::vn_bsubumns = 'bsubumns'
- character(len= *), parameter read_wout_mod::vn_bsubvmns = 'bsubvmns'
- character(len= *), parameter read_wout_mod::vn_bsubsmnc = 'bsubsmnc'
- character(len= *), parameter read_wout_mod::vn_bsupumns = 'bsupumns'
- character(len= *), parameter read_wout_mod::vn_bsupvmns = 'bsupvmns'
- character(len= *), parameter read_wout_mod::vn_bsubumnc_sur = 'bsubumnc_sur'
- character(len= *), parameter read_wout_mod::vn_bsubvmnc_sur = 'bsubvmnc_sur'
- character(len= *), parameter read_wout_mod::vn_bsupumnc_sur = 'bsupumnc_sur'
- character(len= *), parameter read_wout_mod::vn_bsupvmnc_sur = 'bsupvmnc_sur'
- character(len= *), parameter read_wout_mod::vn_bsubumns_sur = 'bsubumns_sur'
- character(len= *), parameter read_wout_mod::vn_bsubvmns_sur = 'bsubvmns_sur'
- character(len= *), parameter read_wout_mod::vn_bsupumns_sur = 'bsupumns_sur'
- character(len= *), parameter read_wout_mod::vn_bsupvmns_sur = 'bsupvmns_sur'
- character(len= *), parameter read_wout_mod::vn_rbc = 'rbc'
- character(len= *), parameter read_wout_mod::vn_zbs = 'zbs'
- character(len= *), parameter read_wout_mod::vn_rbs = 'rbs'
- character(len= *), parameter read_wout_mod::vn_zbc = 'zbc'
- character(len= *), parameter read_wout_mod::vn_potvac = 'potvac'
- character(len= *), parameter read_wout_mod::ln_version = 'VMEC Version'
- character(len= *), parameter read_wout_mod::ln_extension = 'Input file extension'
- character(len= *), parameter read_wout_mod::ln_mgrid = 'MGRID file'
- character(len= *), parameter read_wout_mod::ln_magen = 'Magnetic Energy'
- character(len= *), parameter read_wout_mod::ln_therm = 'Thermal Energy'
- character(len= *), parameter read_wout_mod::ln_gam = 'Gamma'
- character(len= *), parameter read_wout_mod::ln_maxr = 'Maximum R'
- character(len= *), parameter read_wout_mod::ln_minr = 'Minimum R'
- character(len= *), parameter read_wout_mod::ln_maxz = 'Maximum Z'
- character(len= *), parameter read_wout_mod::ln_fp = 'Field Periods'
- character(len= *), parameter read_wout_mod::ln_radnod = 'Radial nodes'
- character(len= *), parameter read_wout_mod::ln_polmod = 'Poloidal modes'
- character(len= *), parameter read_wout_mod::ln_tormod = 'Toroidal modes'
- character(len= *), parameter read_wout_mod::ln_maxmod = 'Fourier modes'
- character(len= *), parameter read_wout_mod::ln_maxmod_nyq = 'Fourier modes (Nyquist)'
- character(len= *), parameter read_wout_mod::ln_maxit = 'Max iterations'

- character(len= *), parameter read_wout_mod::ln_actit = 'Actual iterations'
- character(len= *), parameter read_wout_mod::ln_asym = 'Asymmetry'
- character(len= *), parameter read_wout_mod::ln_recon = 'Reconstruction'
- character(len= *), parameter read_wout_mod::ln_free = 'Free boundary'
- character(len= *), parameter read_wout_mod::ln_error = 'Error flag'
- character(len= *), parameter read_wout_mod::ln_aspect = 'Aspect ratio'
- character(len= *), parameter read_wout_mod::ln_beta = 'Total beta'
- character(len= *), parameter read_wout_mod::ln_pbeta = 'Poloidal beta'
- character(len= *), parameter read_wout_mod::ln_tbeta = 'Toroidal beta'
- character(len= *), parameter read_wout_mod::ln_abeta = 'Beta axis'
- character(len= *), parameter read_wout_mod::ln_b0 = 'RB-t over R axis'
- character(len= *), parameter read_wout_mod::ln_rbt0 = 'RB-t axis'
- character(len= *), parameter read_wout_mod::ln_rbt1 = 'RB-t edge'
- character(len= *), parameter read_wout_mod::ln_sgs = 'Sign jacobian'
- character(len= *), parameter read_wout_mod::ln_lar = 'Ion Larmor radius'
- character(len= *), parameter read_wout_mod::ln_modb = 'avg mod B'
- character(len= *), parameter read_wout_mod::ln_ctor = 'Toroidal current'
- character(len= *), parameter read_wout_mod::ln_amin = 'minor radius'
- character(len= *), parameter read_wout_mod::ln_rmaj = 'major radius'
- character(len= *), parameter read_wout_mod::ln_vol = 'Plasma volume'
- character(len= *), parameter read_wout_mod::ln_mse = 'Number of MSE points'
- character(len= *), parameter read_wout_mod::ln_thom = 'Number of Thompson scattering points'
- character(len= *), parameter read_wout_mod::ln_am = 'Specification parameters for mass(s)'
- character(len= *), parameter read_wout_mod::ln_ac = 'Specification parameters for <J>(s)'
- character(len= *), parameter read_wout_mod::ln_ai = 'Specification parameters for iota(s)'
- character(len= *), parameter read_wout_mod::ln_pmass_type = 'Profile type specifier for mass(s)'
- character(len= *), parameter read_wout_mod::ln_pcurr_type = 'Profile type specifier for <J>(s)'
- character(len= *), parameter read_wout_mod::ln_piota_type = 'Profile type specifier for iota(s)'
- character(len= *), parameter read_wout_mod::ln_am_aux_s = 'Auxiliary-s parameters for mass(s)'
- character(len= *), parameter read_wout_mod::ln_am_aux_f = 'Auxiliary-f parameters for mass(s)'
- character(len= *), parameter read_wout_mod::ln_ac_aux_s = 'Auxiliary-s parameters for <J>(s)'
- character(len= *), parameter read_wout_mod::ln_ac_aux_f = 'Auxiliary-f parameters for <J>(s)'
- character(len= *), parameter read_wout_mod::ln_ai_aux_s = 'Auxiliary-s parameters for iota(s)'
- character(len= *), parameter read_wout_mod::ln_ai_aux_f = 'Auxiliary-f parameters for iota(s)'
- character(len= *), parameter read_wout_mod::ln_pmod = 'Poloidal mode numbers'
- character(len= *), parameter read_wout_mod::ln_tmod = 'Toroidal mode numbers'
- character(len= *), parameter read_wout_mod::ln_pmod_nyq = 'Poloidal mode numbers (Nyquist)'
- character(len= *), parameter read_wout_mod::ln_tmod_nyq = 'Toroidal mode numbers (Nyquist)'
- character(len= *), parameter read_wout_mod::ln_racc = 'raxis (cosnv)'
- character(len= *), parameter read_wout_mod::ln_racs = 'raxis (sinnv)'
- character(len= *), parameter read_wout_mod::ln_zacs = 'zaxis (sinnv)'
- character(len= *), parameter read_wout_mod::ln_zacc = 'zaxis (cosnv)'
- character(len= *), parameter read_wout_mod::ln_iotaf = 'iota on full mesh'
- character(len= *), parameter read_wout_mod::ln_qfact = 'q-factor on full mesh'
- character(len= *), parameter read_wout_mod::ln_presf = 'pressure on full mesh'
- character(len= *), parameter read_wout_mod::ln_phi = 'Toroidal flux on full mesh'
- character(len= *), parameter read_wout_mod::ln_phipf = 'd(phi)/ds: Toroidal flux deriv on full mesh'
- character(len= *), parameter read_wout_mod::ln_chi = 'Poloidal flux on full mesh'
- character(len= *), parameter read_wout_mod::ln_chipf = 'd(chi)/ds: Poloidal flux deriv on full mesh'
- character(len= *), parameter read_wout_mod::ln_jcuru = 'j dot gradu full'
- character(len= *), parameter read_wout_mod::ln_jcurv = 'j dot gradv full'
- character(len= *), parameter read_wout_mod::ln_iotah = 'iota half'
- character(len= *), parameter read_wout_mod::ln_mass = 'mass half'
- character(len= *), parameter read_wout_mod::ln_presh = 'pressure half'
- character(len= *), parameter read_wout_mod::ln_betah = 'beta half'

- character(len= *), parameter `read_wout_mod::ln_buco` = 'bsubu half'
- character(len= *), parameter `read_wout_mod::ln_bvco` = 'bsubv half'
- character(len= *), parameter `read_wout_mod::ln_vp` = 'volume deriv half'
- character(len= *), parameter `read_wout_mod::ln_specw` = 'Spectral width half'
- character(len= *), parameter `read_wout_mod::ln_hiph` = 'tor flux deriv over 2pi half'
- character(len= *), parameter `read_wout_mod::ln_jdotb` = 'J dot B'
- character(len= *), parameter `read_wout_mod::ln_bgrv` = 'B dot grad v'
- character(len= *), parameter `read_wout_mod::ln_merc` = 'Mercier criterion'
- character(len= *), parameter `read_wout_mod::ln_mshear` = 'Shear Mercier'
- character(len= *), parameter `read_wout_mod::ln_mwell` = 'Well Mercier'
- character(len= *), parameter `read_wout_mod::ln_mcurr` = 'Current Mercier'
- character(len= *), parameter `read_wout_mod::ln_mgeo` = 'Geodesic Mercier'
- character(len= *), parameter `read_wout_mod::ln_equif` = 'Average force balance'
- character(len= *), parameter `read_wout_mod::ln_fsq` = 'Residual decay'
- character(len= *), parameter `read_wout_mod::ln_wdot` = 'Wdot decay'
- character(len= *), parameter `read_wout_mod::ln_extcur` = 'External coil currents'
- character(len= *), parameter `read_wout_mod::ln_fsqr` = 'Residual decay - radial'
- character(len= *), parameter `read_wout_mod::ln_fsqz` = 'Residual decay - vertical'
- character(len= *), parameter `read_wout_mod::ln_fsql` = 'Residual decay - hoop'
- character(len= *), parameter `read_wout_mod::ln_ftolv` = 'Residual decay - requested'
- character(len= *), parameter `read_wout_mod::ln_curlab` = 'External current names'
- character(len= *), parameter `read_wout_mod::ln_rmnc` = 'cosmn component of cylindrical R, full mesh'
- character(len= *), parameter `read_wout_mod::ln_zmns` = 'sinmn component of cylindrical Z, full mesh'
- character(len= *), parameter `read_wout_mod::ln_lmns` = 'sinmn component of lambda, half mesh'
- character(len= *), parameter `read_wout_mod::ln_gmnc` = 'cosmn component of [jacobian](#), half mesh'
- character(len= *), parameter `read_wout_mod::ln_bmnc` = 'cosmn component of mod-B, half mesh'
- character(len= *), parameter `read_wout_mod::ln_bsubumnc` = 'cosmn covariant u-component of B, half mesh'
- character(len= *), parameter `read_wout_mod::ln_bsubvmnc` = 'cosmn covariant v-component of B, half mesh'
- character(len= *), parameter `read_wout_mod::ln_bsubsmns` = 'sinmn covariant s-component of B, full mesh'
- character(len= *), parameter `read_wout_mod::ln_bsubumnc_sur` = 'cosmn bsubu of B, [surface](#)'
- character(len= *), parameter `read_wout_mod::ln_bsubvmnc_sur` = 'cosmn bsubv of B, [surface](#)'
- character(len= *), parameter `read_wout_mod::ln_bsupumnc_sur` = 'cosmn bsupu of B, [surface](#)'
- character(len= *), parameter `read_wout_mod::ln_bsupvmnc_sur` = 'cosmn bsupv of B, [surface](#)'
- character(len= *), parameter `read_wout_mod::ln_bsupumnc` = 'BSUPUmnc half'
- character(len= *), parameter `read_wout_mod::ln_bsupvmnc` = 'BSUPVmnc half'
- character(len= *), parameter `read_wout_mod::ln_rmns` = 'sinmn component of cylindrical R, full mesh'
- character(len= *), parameter `read_wout_mod::ln_zmnc` = 'cosmn component of cylindrical Z, full mesh'
- character(len= *), parameter `read_wout_mod::ln_lmnc` = 'cosmn component of lambda, half mesh'
- character(len= *), parameter `read_wout_mod::ln_gmns` = 'sinmn component of [jacobian](#), half mesh'
- character(len= *), parameter `read_wout_mod::ln_bmns` = 'sinmn component of mod-B, half mesh'
- character(len= *), parameter `read_wout_mod::ln_bsubumns` = 'sinmn covariant u-component of B, half mesh'
- character(len= *), parameter `read_wout_mod::ln_bsubvmns` = 'sinmn covariant v-component of B, half mesh'
- character(len= *), parameter `read_wout_mod::ln_bsubsmnc` = 'cosmn covariant s-component of B, full mesh'
- character(len= *), parameter `read_wout_mod::ln_bsubumns_sur` = 'sinmn bsubu of B, [surface](#)'
- character(len= *), parameter `read_wout_mod::ln_bsubvmns_sur` = 'sinmn bsubv of B, [surface](#)'
- character(len= *), parameter `read_wout_mod::ln_bsupumns_sur` = 'sinmn bsupu of B, [surface](#)'
- character(len= *), parameter `read_wout_mod::ln_bsupvmns_sur` = 'sinmn bsupv of B, [surface](#)'
- character(len= *), parameter `read_wout_mod::ln_bsupumns` = 'BSUPUmns half'
- character(len= *), parameter `read_wout_mod::ln_bsupvmns` = 'BSUPVmns half'
- character(len= *), parameter `read_wout_mod::ln_rbc` = 'Initial boundary R cos(mu-nv) coefficients'
- character(len= *), parameter `read_wout_mod::ln_zbs` = 'Initial boundary Z sin(mu-nv) coefficients'
- character(len= *), parameter `read_wout_mod::ln_rbs` = 'Initial boundary R sin(mu-nv) coefficients'
- character(len= *), parameter `read_wout_mod::ln_zbc` = 'Initial boundary Z cos(mu-nv) coefficients'
- character(len= *), parameter `read_wout_mod::ln_potvac` = 'Vacuum Potential on Boundary'
- integer `read_wout_mod::nfp`

- integer `read_wout_mod::ns`
- integer `read_wout_mod::mpol`
- integer `read_wout_mod::ntor`
- integer `read_wout_mod::mnmax`
- integer `read_wout_mod::mnmax_nyq`
- integer `read_wout_mod::itfsq`
- integer `read_wout_mod::niter`
- integer `read_wout_mod::iasym`
- integer `read_wout_mod::ierr_vmec`
- integer `read_wout_mod::imse`
- integer `read_wout_mod::itse`
- integer `read_wout_mod::nstore_seq`
- integer `read_wout_mod::isnodes`
- integer `read_wout_mod::ipnodes`
- integer `read_wout_mod::imatch_phiedge`
- integer `read_wout_mod::isigng`
- integer `read_wout_mod::mnyq`
- integer `read_wout_mod::nnyq`
- integer `read_wout_mod::ntmax`
- real(rprec) `read_wout_mod::wb`
- real(rprec) `read_wout_mod::wp`
- real(rprec) `read_wout_mod::gamma`
- real(rprec) `read_wout_mod::pfac`
- real(rprec) `read_wout_mod::rmax_surf`
- real(rprec) `read_wout_mod::rmin_surf`
- real(rprec) `read_wout_mod::zmax_surf`
- real(rprec) `read_wout_mod::aspect`
- real(rprec) `read_wout_mod::betatot`
- real(rprec) `read_wout_mod::betapol`
- real(rprec) `read_wout_mod::betator`
- real(rprec) `read_wout_mod::betaxis`
- real(rprec) `read_wout_mod::b0`
- real(rprec) `read_wout_mod::tswgt`
- real(rprec) `read_wout_mod::msewgt`
- real(rprec) `read_wout_mod::flmwgt`
- real(rprec) `read_wout_mod::bcwgt`
- real(rprec) `read_wout_mod::phidiam`
- real(rprec) `read_wout_mod::version_`
- real(rprec) `read_wout_mod::delphid`
- real(rprec) `read_wout_mod::ionlarmor`
- real(rprec) `read_wout_mod::volavgb`
- real(rprec) `read_wout_mod::fsql`
- real(rprec) `read_wout_mod::fsqr`
- real(rprec) `read_wout_mod::fsqz`
- real(rprec) `read_wout_mod::ftolv`
- real(rprec) `read_wout_mod::aminor`
- real(rprec) `read_wout_mod::rmajor`
- real(rprec) `read_wout_mod::volume`
- real(rprec) `read_wout_mod::rbtor`
- real(rprec) `read_wout_mod::rbtor0`
- real(rprec) `read_wout_mod::itor`
- real(rprec) `read_wout_mod::machsq`
- real(rprec), dimension(:,:,:), allocatable `read_wout_mod::rzi_local`
- real(rprec), dimension(:,:), allocatable `read_wout_mod::rmnc`
- real(rprec), dimension(:,:), allocatable `read_wout_mod::zmns`

- `real(rprec), dimension(:, :), allocatable read_wout_mod::lmns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::rmns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::zmnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::lmnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bmnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::gmnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsubumnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsubvmnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsubsmns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsupumnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsupvmnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::curvmnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::currumnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bbc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::raxis`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::zaxis`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bmns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::gmns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsubumns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsubvmns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsubsmnc`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsupumns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::bsupvmns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::currumns`
- `real(rprec), dimension(:, :), allocatable read_wout_mod::curvmns`
- `real(rprec), dimension(:), allocatable read_wout_mod::iotas`
- `real(rprec), dimension(:), allocatable read_wout_mod::iotaf`
- `real(rprec), dimension(:), allocatable read_wout_mod::presf`
- `real(rprec), dimension(:), allocatable read_wout_mod::phipf`
- `real(rprec), dimension(:), allocatable read_wout_mod::mass`
- `real(rprec), dimension(:), allocatable read_wout_mod::pres`
- `real(rprec), dimension(:), allocatable read_wout_mod::beta_vol`
- `real(rprec), dimension(:), allocatable read_wout_mod::xm`
- `real(rprec), dimension(:), allocatable read_wout_mod::xn`
- `real(rprec), dimension(:), allocatable read_wout_mod::qfact`
- `real(rprec), dimension(:), allocatable read_wout_mod::chipf`
- `real(rprec), dimension(:), allocatable read_wout_mod::phi`
- `real(rprec), dimension(:), allocatable read_wout_mod::chi`
- `real(rprec), dimension(:), allocatable read_wout_mod::xm_nyq`
- `real(rprec), dimension(:), allocatable read_wout_mod::xn_nyq`
- `real(rprec), dimension(:), allocatable read_wout_mod::phip`
- `real(rprec), dimension(:), allocatable read_wout_mod::buco`
- `real(rprec), dimension(:), allocatable read_wout_mod::bvco`
- `real(rprec), dimension(:), allocatable read_wout_mod::vp`
- `real(rprec), dimension(:), allocatable read_wout_mod::overr`
- `real(rprec), dimension(:), allocatable read_wout_mod::jcuru`
- `real(rprec), dimension(:), allocatable read_wout_mod::jcurv`
- `real(rprec), dimension(:), allocatable read_wout_mod::specw`
- `real(rprec), dimension(:), allocatable read_wout_mod::jdotb`
- `real(rprec), dimension(:), allocatable read_wout_mod::bdotgradv`
- `real(rprec), dimension(:), allocatable read_wout_mod::fsqt`
- `real(rprec), dimension(:), allocatable read_wout_mod::wdot`
- `real(rprec), dimension(:), allocatable read_wout_mod::am`
- `real(rprec), dimension(:), allocatable read_wout_mod::ac`
- `real(rprec), dimension(:), allocatable read_wout_mod::ai`

- real(rprec), dimension(:), allocatable [read_wout_mod::am_aux_s](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::am_aux_f](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::ac_aux_s](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::ac_aux_f](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::ai_aux_s](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::ai_aux_f](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::dmerc](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::dshear](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::dwell](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::dcurr](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::dgeod](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::equif](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::extcur](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::sknots](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::ystark](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::y2stark](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::pknots](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::ythom](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::y2thom](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::anglemse](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::rmid](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::qmid](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::shear](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::presmid](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::alfa](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::curmid](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::rstark](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::qmeas](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::datastark](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::rthom](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::datathom](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::dsiobt](#)
- real(rprec), dimension(:), allocatable [read_wout_mod::potvac](#)
- logical [read_wout_mod::lasym](#)
- logical [read_wout_mod::lthreed](#)
- logical [read_wout_mod::lwout_opened](#) = .false.
- character [read_wout_mod::mgrid_file](#)
- character [read_wout_mod::input_extension](#)
- character [read_wout_mod::pmass_type](#)
- character [read_wout_mod::pcurr_type](#)
- character [read_wout_mod::piota_type](#)

7.88.1 Detailed Description

Reading of `wout` VMEC output file.

7.89 src/readin.f90 File Reference

Read the input file.

Functions/Subroutines

- subroutine `readin` (`input_file`, `ier_flag`)
Read the input file.

7.89.1 Detailed Description

Read the input file.

7.89.2 Function/Subroutine Documentation

7.89.2.1 `readin()`

```
subroutine readin (
    character(len=*) input_file,
    integer, intent(inout) ier_flag )
```

Read the input file.

Parameters

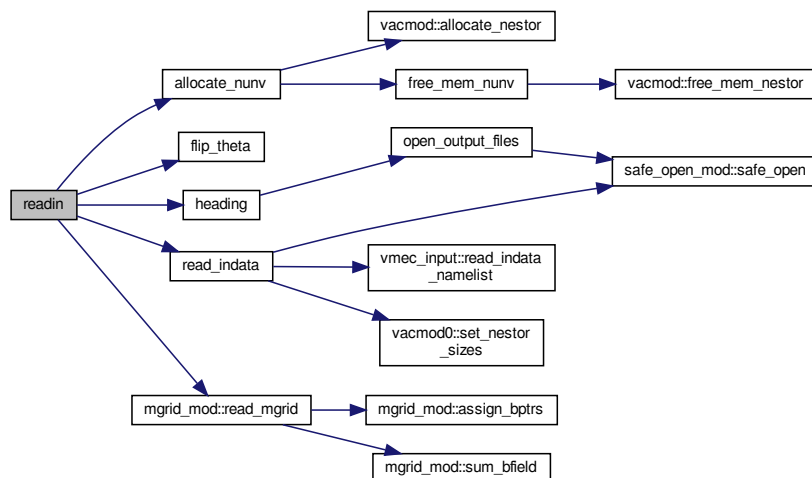
<i>input_file</i>	input file to read from
<i>ier_flag</i>	error flag

Definition at line 8 of file `readin.f90`.

References `allocate_nunv()`, `mgrid_mod::curlabel`, `vmec_main::currv`, `flip_theta()`, `heading()`, `vmec_main::lconm1`, `vmec_main::lflip`, `vmec_main::lthreed`, `vmec_main::multi_ns_grid`, `mgrid_mod::nextcur`, `vmec_params::norm_↵`
`term_flag`, `mgrid_mod::np0b`, `mgrid_mod::nr0b`, `mgrid_mod::nz0b`, `vmec_params::rcc`, `vmec_params::rcs`, `read_↵`
`indata()`, `mgrid_mod::read_mgrid()`, `mgrid_mod::rmaxb`, `mgrid_mod::rminb`, `vmec_main::rmn_bdy`, `vmec_params↵`
`::rsc`, `vmec_params::rss`, `vmec_params::signgs`, `vmec_params::zcc`, `vmec_params::zcs`, `mgrid_mod::zmaxb`,
`mgrid_mod::zminb`, `vmec_main::zmn_bdy`, `vmec_params::zsc`, and `vmec_params::zss`.

Referenced by `vmec()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.90 src/reset_params.f90 File Reference

Functions/Subroutines

- subroutine [reset_params](#)

7.90.1 Function/Subroutine Documentation

7.90.1.1 reset_params()

```
subroutine reset_params
```

m=1 constraint (=t: apply correct, polar constraint; =f, apply approx. constraint)

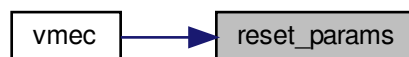
Assume scaled mode; read in from mgrid in free-bdy mode

Definition at line 2 of file reset_params.f90.

References vmec_input::delt, vmec_main::delt0r, vmec_main::fsq, vmec_main::fsqr, vmec_main::fsqz, vmec_main::ftolv, vmec_main::ijacob, vmec_main::irst, vmec_main::iter1, vmec_main::iter2, vmec_main::ivac, vmec_main::lconm1, mgrid_mod::mgrid_mode, mgrid_mod::nextcur, vmec_main::res0, vmec_main::vacuum_calls, vmec_main::z00, and stel_constants::zero.

Referenced by vmec().

Here is the caller graph for this function:



7.91 src/residue.f90 File Reference

Functions/Subroutines

- subroutine [residue](#) (gcr, gcx, gcl)
- subroutine [constrain_m1](#) (gcr, gcx)
- subroutine [scale_m1](#) (gcr, gcx)

7.91.1 Function/Subroutine Documentation

7.91.1.1 constrain_m1()

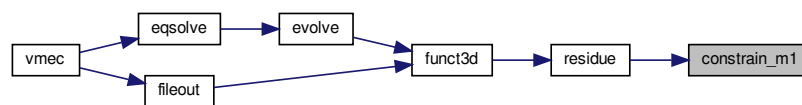
```
subroutine constrain_m1 (
    real(dp), dimension(ns,0:ntor), intent(inout) gcr,
    real(dp), dimension(ns,0:ntor), intent(inout) gcz )
```

Definition at line 81 of file residue.f90.

References vmec_main::fsqz, vmec_main::iter2, and vmec_main::lconm1.

Referenced by residue().

Here is the caller graph for this function:



7.91.1.2 residue()

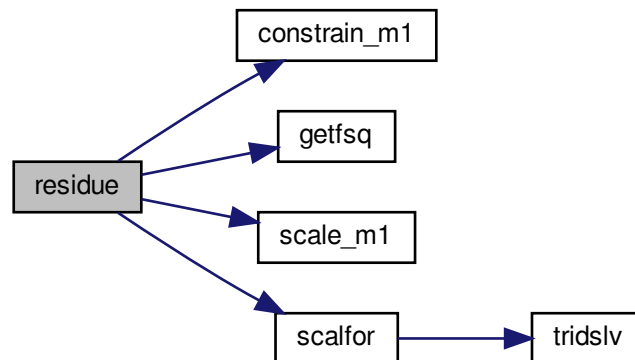
```
subroutine residue (
    real(rprec), dimension(ns,0:ntor,0:mpoll,ntmax), intent(inout) gcr,
    real(rprec), dimension(ns,0:ntor,0:mpoll,ntmax), intent(inout) gcz,
    real(rprec), dimension(ns,0:ntor,0:mpoll,ntmax), intent(inout) gcl )
```

Definition at line 2 of file residue.f90.

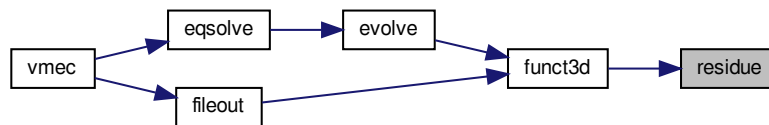
References vmec_main::ard, vmec_main::arm, vmec_main::azd, vmec_main::azm, vmec_main::brd, vmec_main::brm, vmec_main::bzd, vmec_main::bzm, constrain_m1(), vmec_main::crd, vmec_main::faclam, vmec_main::fedge, vmec_main::fnorm, vmec_main::fnorm1, vmec_main::fnorml, vmec_main::fsql, vmec_main::fsql1, vmec_main::fsqr, vmec_main::fsqr1, vmec_main::fsqz, vmec_main::fsqz1, getfsq(), vmec_main::hs, vmec_main::iter1, vmec_main::iter2, vmec_main::lthreed, vmec_params::meven, vmec_params::modd, vmec_params::ntmax, vmec_main::r0scale, vmec_params::rsc, vmec_params::rss, scale_m1(), scalfor(), vmec_params::zcc, and vmec_params::zcs.

Referenced by funct3d().

Here is the call graph for this function:



Here is the caller graph for this function:



7.91.1.3 scale_m1()

```

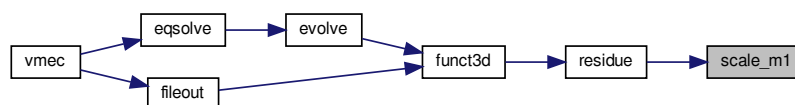
subroutine scale_m1 (
    real(rprec), dimension(ns,0:ntor), intent(inout) gcr,
    real(rprec), dimension(ns,0:ntor), intent(inout) gcx )
  
```

Definition at line 105 of file residue.f90.

References vmec_main::ard, vmec_main::azd, vmec_main::brd, vmec_main::bzd, and vmec_main::lconm1.

Referenced by residue().

Here is the caller graph for this function:



7.92 src/restart_iter.f90 File Reference

Functions/Subroutines

- subroutine [restart_iter](#) (time_step)

7.92.1 Function/Subroutine Documentation

7.92.1.1 restart_iter()

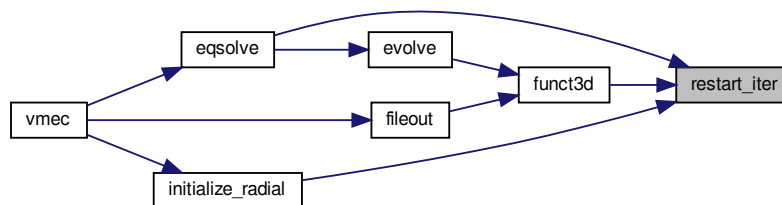
```
subroutine restart_iter (
    real(rprec), intent(inout) time_step )
```

Definition at line 2 of file restart_iter.f90.

References vmec_main::ijacob, vmec_main::irst, vmec_main::iter1, vmec_main::iter2, vmec_main::neqs, xstuff::xc, xstuff::xcdot, and xstuff::xstore.

Referenced by eqsolve(), funct3d(), and initialize_radial().

Here is the caller graph for this function:



7.93 src/safe_open_mod.f File Reference

Modules

- module [safe_open_mod](#)

Functions/Subroutines

- subroutine [safe_open_mod::safe_open](#) (iunit, istat, filename, filestat, fileform, record_in, access_in, delim_in)

7.94 src/scalfor.f90 File Reference

Functions/Subroutines

- subroutine [scalfor](#) (gcx, axm, bxm, axd, bxd, cx, iflag)

7.94.1 Function/Subroutine Documentation

7.94.1.1 scalfor()

```
subroutine scalfor (
    real(rprec), dimension(ns,0:ntor,0:mpoll,ntmax), intent(inout) gcx,
    real(rprec), dimension(ns+1,2), intent(in) axm,
    real(rprec), dimension(ns+1,2), intent(in) bxm,
    real(rprec), dimension(ns+1,2), intent(in) axd,
    real(rprec), dimension(ns+1,2), intent(in) bxd,
    real(rprec), dimension(ns), intent(in) cx,
    integer, intent(in) iflag )
```

Definition at line 2 of file scalfor.f90.

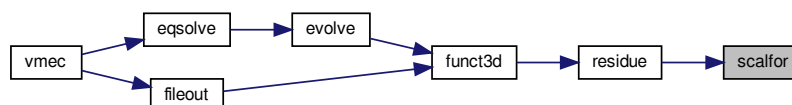
References `vmec_main::hs`, `vmec_main::ivac`, `vmec_params::jmin2`, `vmec_dim::ns`, `vmec_params::ntmax`, and `tridslv()`.

Referenced by `residue()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.95 src/solver.f90 File Reference

Functions/Subroutines

- subroutine [solver](#) (amat, b, m, nrhs, info)

7.95.1 Function/Subroutine Documentation

7.95.1.1 solver()

```

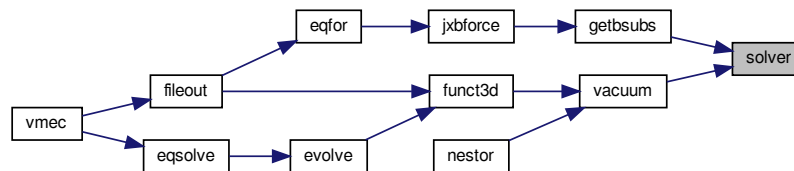
subroutine solver (
    real(rprec), dimension(m,m), intent(inout) amat,
    real(rprec), dimension(m,nrhs), intent(inout) b,
    integer, intent(in) m,
    integer, intent(in) nrhs,
    integer, intent(out) info )

```

Definition at line 2 of file solver.f90.

Referenced by [getbsubs\(\)](#), and [vacuum\(\)](#).

Here is the caller graph for this function:



7.96 src/spectrum.f90 File Reference

Functions/Subroutines

- subroutine [spectrum](#) (rmn, zmn)

7.96.1 Function/Subroutine Documentation

7.96.1.1 spectrum()

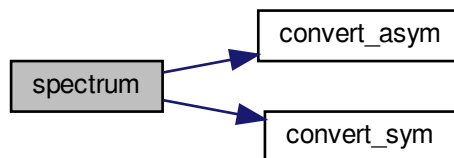
```
subroutine spectrum (
    real(rprec), dimension(ns,0:ntor,0:mpoll,ntmax), intent(inout) rmn,
    real(rprec), dimension(ns,0:ntor,0:mpoll,ntmax), intent(inout) zmn )
```

Definition at line 2 of file spectrum.f90.

References `convert_asym()`, `convert_sym()`, `vmec_main::lthreed`, `vmec_params::mscale`, `vmec_params::nscale`, `vmec_params::ntmax`, `vmec_params::rsc`, `vmec_params::rss`, `vmec_main::specw`, `vmec_main::xmpq`, `vmec_params::zcc`, and `vmec_params::zcs`.

Referenced by `printout()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.97 src/spline_akima.f File Reference

Functions/Subroutines

- subroutine `spline_akima` (`x`, `y`, `xx`, `yy`, `npts`, `iflag`)

7.97.1 Function/Subroutine Documentation

7.97.1.1 spline_akima()

```

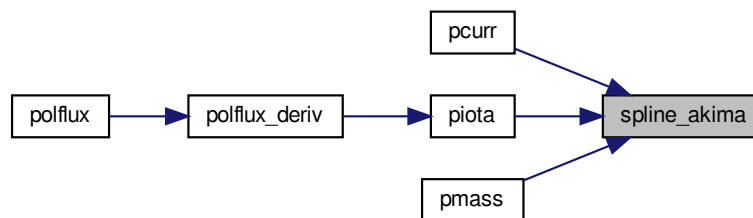
subroutine spline_akima (
    real(rprec), intent(in) x,
    real(rprec), intent(out) y,
    real(rprec), dimension(npts), intent(in) xx,
    real(rprec), dimension(npts), intent(in) yy,
    integer, intent(in) npts,
    integer, intent(inout) iflag )

```

Definition at line 2 of file spline_akima.f.

Referenced by pcurr(), piota(), and pmass().

Here is the caller graph for this function:



7.98 src/spline_akima_int.f File Reference

Functions/Subroutines

- subroutine [spline_akima_int](#) (x, y, xx, yy, npts, iflag)

7.98.1 Function/Subroutine Documentation

7.98.1.1 spline_akima_int()

```

subroutine spline_akima_int (
    real(rprec), intent(in) x,
    real(rprec), intent(out) y,
    real(rprec), dimension(npts), intent(in) xx,
    real(rprec), dimension(npts), intent(in) yy,
    integer, intent(in) npts,
    integer, intent(inout) iflag )

```

Definition at line 2 of file spline_akima_int.f.

Referenced by pcurr().

Here is the caller graph for this function:



7.99 src/spline_cubic.f File Reference

Functions/Subroutines

- subroutine [spline_cubic](#) (x, y, xx, yy, n, iflag)
- subroutine [spline_nr](#) (x, y, n, yp1, ypn, y2)
- subroutine [splint_nr](#) (xa, ya, y2a, n, x, y)

7.99.1 Function/Subroutine Documentation

7.99.1.1 spline_cubic()

```

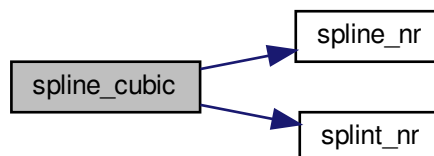
subroutine spline_cubic (
    real(rprec), intent(in) x,
    real(rprec), intent(out) y,
    real(rprec), dimension(n), intent(in) xx,
    real(rprec), dimension(n), intent(in) yy,
    integer, intent(in) n,
    integer, intent(inout) iflag )
  
```

Definition at line 2 of file spline_cubic.f.

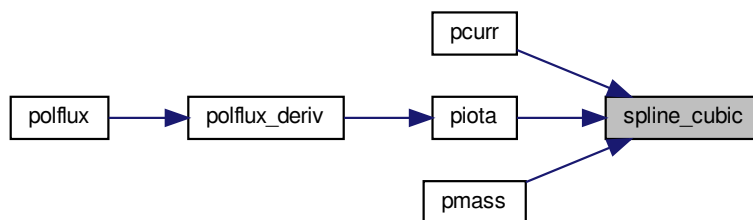
References [spline_nr\(\)](#), and [splint_nr\(\)](#).

Referenced by [pcurr\(\)](#), [piota\(\)](#), and [pmass\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



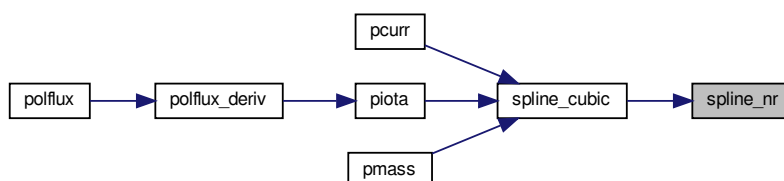
7.99.1.2 spline_nr()

```
subroutine spline_cubic::spline_nr (  
    real(rprec), dimension(n), intent(in) x,  
    real(rprec), dimension(n), intent(in) y,  
    integer, intent(in) n,  
    real(rprec), intent(in) yp1,  
    real(rprec), intent(in) ypn,  
    real(rprec), dimension(n), intent(out) y2 )
```

Definition at line 56 of file `spline_cubic.f`.

Referenced by `spline_cubic()`.

Here is the caller graph for this function:



7.99.1.3 splint_nr()

```

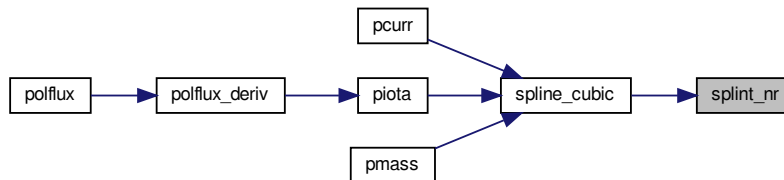
subroutine spline_cubic::splint_nr (
    real(rprec), dimension(n), intent(in) xa,
    real(rprec), dimension(n), intent(in) ya,
    real(rprec), dimension(n), intent(in) y2a,
    integer, intent(in) n,
    real(rprec), intent(in) x,
    real(rprec), intent(out) y )

```

Definition at line 105 of file spline_cubic.f.

Referenced by spline_cubic().

Here is the caller graph for this function:



7.100 src/spline_cubic_int.f File Reference

Functions/Subroutines

- subroutine [spline_cubic_int](#) (x, y, xx, yy, n, iflag)
- subroutine [spline_int](#) (x, y, n, yp1, ypn, y2)
- subroutine [splint_int](#) (xa, ya, y2a, n, x, y)

7.100.1 Function/Subroutine Documentation

7.100.1.1 spline_cubic_int()

```

subroutine spline_cubic_int (
    real(rprec), intent(in) x,
    real(rprec), intent(out) y,
    real(rprec), dimension(n), intent(in) xx,
    real(rprec), dimension(n), intent(in) yy,
    integer, intent(in) n,
    integer, intent(inout) iflag )

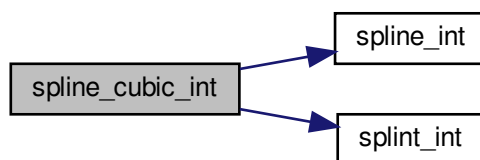
```

Definition at line 2 of file spline_cubic_int.f.

References spline_int(), and splint_int().

Referenced by pcurr().

Here is the call graph for this function:



Here is the caller graph for this function:



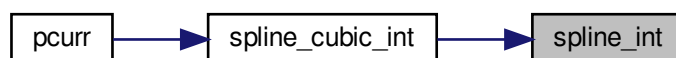
7.100.1.2 spline_int()

```
subroutine spline_cubic_int::spline_int (
    real(rprec), dimension(n), intent(in) x,
    real(rprec), dimension(n), intent(in) y,
    integer, intent(in) n,
    real(rprec), intent(in) yp1,
    real(rprec), intent(in) ypn,
    real(rprec), dimension(n), intent(out) y2 )
```

Definition at line 56 of file spline_cubic_int.f.

Referenced by spline_cubic_int().

Here is the caller graph for this function:



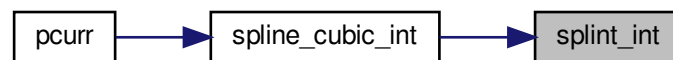
7.100.1.3 splint_int()

```
subroutine spline_cubic_int::splint_int (
    real(rprec), dimension(n), intent(in) xa,
    real(rprec), dimension(n), intent(in) ya,
    real(rprec), dimension(n), intent(in) y2a,
    integer, intent(in) n,
    real(rprec), intent(in) x,
    real(rprec), intent(out) y )
```

Definition at line 105 of file spline_cubic_int.f.

Referenced by spline_cubic_int().

Here is the caller graph for this function:



7.101 src/symforce.f90 File Reference

Functions/Subroutines

- subroutine [symforce](#) (ars, brs, crs, azs, bzs, czs, bls, cls, rcs, zcs, ara, bra, cra, aza, bza, cza, bla, cla, rca, zca)
- subroutine [symoutput](#) (bsq, gsqr, bsubu, bsubv, bsupu, bsupv, bsubs, bsqa, gsqrta, bsubua, bsubva, bsupua, bsupva, bsubsa)

7.101.1 Function/Subroutine Documentation

7.101.1.1 symforce()

```

subroutine symforce (
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) ars,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) brs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) crs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) azs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) bzs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) czs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) bls,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) cls,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) rcs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) zcs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) ara,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) bra,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) cra,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) aza,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) bza,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) cza,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) bla,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) cla,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) rca,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) zca )

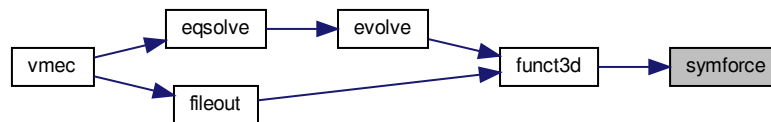
```

Definition at line 2 of file symforce.f90.

References vmec_main::ireflect, and vmec_main::lthread.

Referenced by funct3d().

Here is the caller graph for this function:



7.101.1.2 symoutput()

```

subroutine symoutput (
    real(rprec), dimension(ns*nzeta,ntheta3), intent(inout) bsq,
    real(rprec), dimension(ns*nzeta,ntheta3), intent(inout) gsqr,
    real(rprec), dimension(ns*nzeta,ntheta3), intent(inout) bsubu,
    real(rprec), dimension(ns*nzeta,ntheta3), intent(inout) bsubv,
    real(rprec), dimension(ns*nzeta,ntheta3), intent(inout) bsupu,
    real(rprec), dimension(ns*nzeta,ntheta3), intent(inout) bsupv,
    real(rprec), dimension(ns*nzeta,ntheta3), intent(inout) bsubs,

```

```

real(rprec), dimension(ns*nzeta,ntheta3), intent(out) bsqa,
real(rprec), dimension(ns*nzeta,ntheta3), intent(out) gsqrta,
real(rprec), dimension(ns*nzeta,ntheta3), intent(out) bsubua,
real(rprec), dimension(ns*nzeta,ntheta3), intent(out) bsubva,
real(rprec), dimension(ns*nzeta,ntheta3), intent(out) bsupua,
real(rprec), dimension(ns*nzeta,ntheta3), intent(out) bsupva,
real(rprec), dimension(ns*nzeta,ntheta3), intent(out) bsubsa )

```

Definition at line 81 of file symforce.f90.

References vmec_main::ireflect.

Referenced by wrout().

Here is the caller graph for this function:



7.102 src/symrzl.f90 File Reference

Functions/Subroutines

- subroutine [symrzl](#) (r1s, rus, rvs, z1s, zus, zvs, lus, lvs, rcons, zcons, r1a, rua, rva, z1a, zua, zva, lua, lva, rcona, zcona)

7.102.1 Function/Subroutine Documentation

7.102.1.1 symrzl()

```

subroutine symrzl (
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) r1s,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) rus,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) rvs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) z1s,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) zus,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) zvs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) lus,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) lvs,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) rcons,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(inout) zcons,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) r1a,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) rua,

```

```

real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) rva,
real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) zla,
real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) zua,
real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) zva,
real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) lua,
real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) lva,
real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) rcona,
real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) zcona )

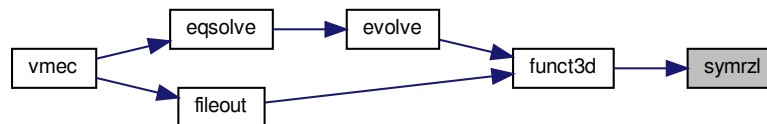
```

Definition at line 2 of file symrzl.f90.

References vmec_main::ireflect, and vmec_main::lthread.

Referenced by funct3d().

Here is the caller graph for this function:



7.103 src/tolower.f90 File Reference

Functions/Subroutines

- subroutine [tolower](#) (string)

7.103.1 Function/Subroutine Documentation

7.103.1.1 tolower()

```

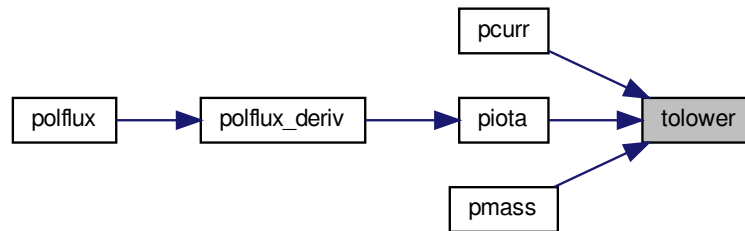
subroutine tolower (
    character (len=*), intent(inout) string )

```

Definition at line 2 of file tolower.f90.

Referenced by pcurr(), piota(), and pmass().

Here is the caller graph for this function:



7.104 src/tomnsp.f90 File Reference

Functions/Subroutines

- subroutine [tomnsp](#) (frzl_array, armn, brmn, crmn, azmn, bzm, czmn, blmn, clmn, arcon, azcon)
- subroutine [tomnspa](#) (frzl_array, armn, brmn, crmn, azmn, bzm, czmn, blmn, clmn, arcon, azcon)

7.104.1 Function/Subroutine Documentation

7.104.1.1 tomnspa()

```

subroutine tomnspa (
    real(rprec), dimension(ns,0:ntor,0:mpoll,3*ntmax), intent(inout), target frzl_←
array,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) armn,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) brmn,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) crmn,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) azmn,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) bzm,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) czmn,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) blmn,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) clmn,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) arcon,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(in) azcon )

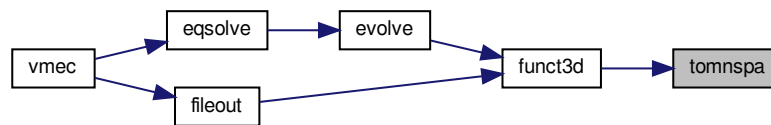
```

Definition at line 120 of file tomnsp.f90.

References [vmec_main::ivac](#), [vmec_params::jlam](#), [vmec_params::jmin2](#), [vmec_main::lthreed](#), [vmec_params::ntmax](#), [vmec_params::rcs](#), [vmec_params::rsc](#), [vmec_main::xmpq](#), [vmec_params::zcc](#), and [vmec_params::zss](#).

Referenced by [funct3d\(\)](#).

Here is the caller graph for this function:



7.104.1.2 tomnsp()

```

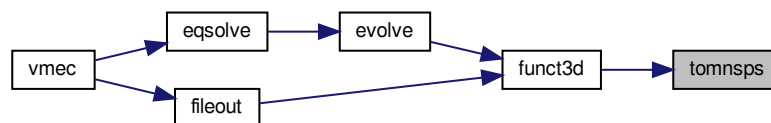
subroutine tomnsp (
    real(rprec), dimension(ns,0:ntor,0:mpoll,3*ntmax), intent(out), target frzl_array,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) armn,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) brmn,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) crmn,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) azmn,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) bzm,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) czmn,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) blmn,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) clmn,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) arcon,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(in) azcon )
  
```

Definition at line 2 of file tomnsp.f90.

References vmec_main::ivac, vmec_params::jlam, vmec_params::jmin2, vmec_main::lthreed, vmec_params::ntmax, vmec_params::rcc, vmec_params::rss, vmec_main::xmpq, vmec_params::zcs, and vmec_params::zsc.

Referenced by funct3d().

Here is the caller graph for this function:



7.105 src/totzsp.f90 File Reference

Functions/Subroutines

- subroutine [totzsp](#) (rzl_array, r11, ru1, rv1, z11, zu1, zv1, lu1, lv1, rcn1, zcn1)
- subroutine [convert_sym](#) (rmnss, zmncs)
- subroutine [totzspa](#) (rzl_array, r11, ru1, rv1, z11, zu1, zv1, lu1, lv1, rcn1, zcn1)
- subroutine [convert_asym](#) (rmnsc, zmnc)

7.105.1 Function/Subroutine Documentation

7.105.1.1 `convert_asym()`

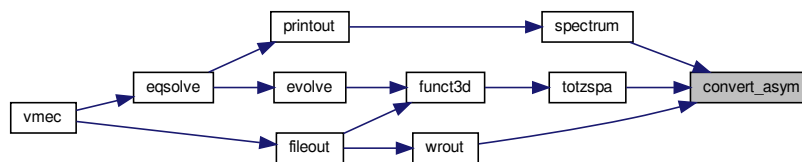
```
subroutine convert_asym (
    real(rprec), dimension(ns,0:ntor,0:mpoll), intent(inout) rmnsc,
    real(rprec), dimension(ns,0:ntor,0:mpoll), intent(inout) zmncc )
```

Definition at line 310 of file `totzsp.f90`.

References `vmec_main::lconm1`, and `stel_kinds::rprec`.

Referenced by `spectrum()`, `totzspa()`, and `wrout()`.

Here is the caller graph for this function:



7.105.1.2 `convert_sym()`

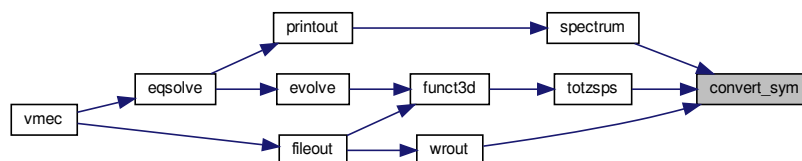
```
subroutine convert_sym (
    real(rprec), dimension(ns,0:ntor,0:mpoll), intent(inout) rmnss,
    real(rprec), dimension(ns,0:ntor,0:mpoll), intent(inout) zmnccs )
```

Definition at line 170 of file `totzsp.f90`.

References `vmec_main::lconm1`, `vmec_params::m1`, and `stel_kinds::rprec`.

Referenced by `spectrum()`, `totzsps()`, and `wrout()`.

Here is the caller graph for this function:



7.105.1.3 totzspa()

```

subroutine totzspa (
    real(rprec), dimension(ns,0:ntor,0:mpol1,3*ntmax), intent(inout), target rzl_↵
array,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) r11,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) ru1,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) rv1,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) z11,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) zu1,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) zv1,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) lu1,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) lv1,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) rcn1,
    real(rprec), dimension(ns*nzeta,ntheta3,0:1), intent(out) zcn1 )

```

Definition at line 193 of file totzsp.f90.

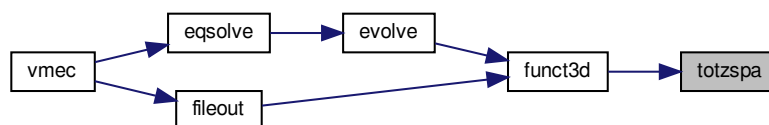
References `convert_asym()`, `vmec_params::jlam`, `vmec_params::jmin1`, `vmec_main::lthreed`, `vmec_params↵`
`::m0`, `vmec_params::m1`, `vmec_params::n0`, `vmec_params::ntmax`, `vmec_params::rcs`, `stel_kinds::rprec`, `vmec_↵`
`params::rsc`, `vmec_main::xmpq`, `vmec_params::zcc`, and `vmec_params::zss`.

Referenced by `funct3d()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.105.1.4 totzsps()

```

subroutine totzsps (
    real(rprec), dimension(ns,0:ntor,0:mpol1,3*ntmax), intent(inout), target rzl_↵
array,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) r11,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) ru1,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) rv1,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) z11,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) zu1,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) zv1,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) lu1,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) lv1,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) rcn1,
    real(rprec), dimension(ns*nzeta*ntheta3,0:1), intent(out) zcn1 )

```

Parameters

out	<i>r11</i>	R
out	<i>ru1</i>	dR/dTheta
out	<i>rv1</i>	dR/dZeta
out	<i>z11</i>	Z
out	<i>zu1</i>	dZ/dTheta
out	<i>zv1</i>	dZ/dZeta
out	<i>lu1</i>	dLambda/dTheta
out	<i>lv1</i>	-dLambda/dZeta
out	<i>rcn1</i>	TODO: what is this?
out	<i>zcn1</i>	TODO: what is this?

Definition at line 2 of file totzsp.f90.

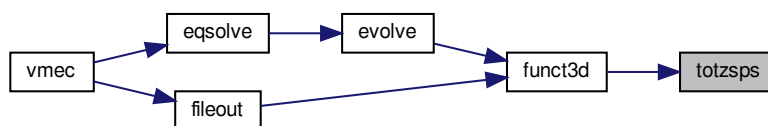
References `convert_sym()`, `vmec_params::jlam`, `vmec_params::jmin1`, `vmec_main::lthreed`, `vmec_params::m0`, `vmec_params::m1`, `vmec_params::n0`, `vmec_params::ntmax`, `vmec_params::rcc`, `stel_kinds::rprec`, `vmec_↵`
`params::rss`, `vmec_main::xmpq`, `vmec_params::zcs`, and `vmec_params::zsc`.

Referenced by `funct3d()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.106 src/tridslv.f90 File Reference

Functions/Subroutines

- subroutine [tridslv](#) (a, d, b, c, jmin, jmax, mnd1, ns, nrhs)

7.106.1 Function/Subroutine Documentation

7.106.1.1 tridslv()

```

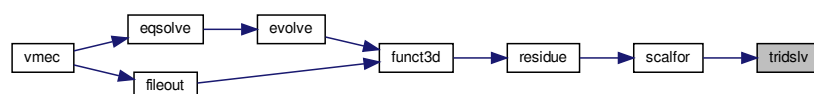
subroutine tridslv (
    real(rprec), dimension(ns,0:mnd1) a,
    real(rprec), dimension(ns,0:mnd1) d,
    real(rprec), dimension(ns,0:mnd1) b,
    real(rprec), dimension(ns,0:mnd1, nrhs), intent(inout) c,
    integer, dimension(0:mnd1), intent(in) jmin,
    integer, intent(in) jmax,
    integer, intent(in) mnd1,
    integer, intent(in) ns,
    integer, intent(in) nrhs )

```

Definition at line 2 of file tridslv.f90.

Referenced by [scalfor\(\)](#).

Here is the caller graph for this function:



7.107 src/vmec.f90 File Reference

Main program of VMEC.

Functions/Subroutines

- program [xvmec](#)
Main program of VMEC.
- subroutine [vmec](#)
Main driver for VMEC.

7.107.1 Detailed Description

Main program of VMEC.

7.107.2 Function/Subroutine Documentation

7.107.2.1 vmec()

```
subroutine vmec
```

Main driver for VMEC.

Definition at line 12 of file vmec.f90.

References `vmec_params::bad_jacobian_flag`, `vmec_input::delt`, `vmec_main::delt0r`, `eqsolve()`, `fileout()`, `fixaray()`, `free_mem_funct3d()`, `free_mem_ns()`, `free_mem_nunv()`, `mgrid_mod::free_mgrid()`, `vmec_input::ftol_array`, `vmec↵_main::ftolv`, `vmec_main::iequi`, `initialize_radial()`, `vmec_input::input_extension`, `vmec_main::ivac`, `vmec_input↵::lfreeb`, `vmec_params::mscale`, `vmec_main::multi_ns_grid`, `vmec_input::niter_array`, `vmec_main::niterv`, `vmec↵_params::norm_term_flag`, `vmec_input::ns_array`, `vmec_params::nscale`, `vparams::nthreed`, `readin()`, `reset_↵params()`, `vmec_params::successful_term_flag`, and `vmec_params::uminus`.

```
program xvmec
```

Definition at line 5 of file vmec.f90.

7.108 src/wrout.f90 File Reference

Functions/Subroutines

- subroutine [wrout](#) (bsq, gsqrt, bsubu, bsubv, bsubs, bsupv, bsupu, rzl_array, gc_array, ier_flag)

7.108.1 Function/Subroutine Documentation

7.108.1.1 wrout()

```

subroutine wrout (
    real(rprec), dimension(ns,nznt), intent(inout) bsq,
    real(rprec), dimension(ns,nznt), intent(inout) gsqrt,
    real(rprec), dimension(ns,nznt), intent(inout) bsubu,
    real(rprec), dimension(ns,nznt), intent(inout) bsubv,
    real(rprec), dimension(ns,nznt), intent(inout) bsubs,
    real(rprec), dimension(ns,nznt), intent(inout) bsupv,
    real(rprec), dimension(ns,nznt), intent(inout) bsupu,
    real(rprec), dimension(mnmax,ns,3*max(ntmax/2,1)), intent(inout), target rzl_↵
array,
    real(rprec), dimension(mnmax,ns,3*max(ntmax/2,1)), intent(inout), target gc_array,
    integer, intent(in) ier_flag )

```

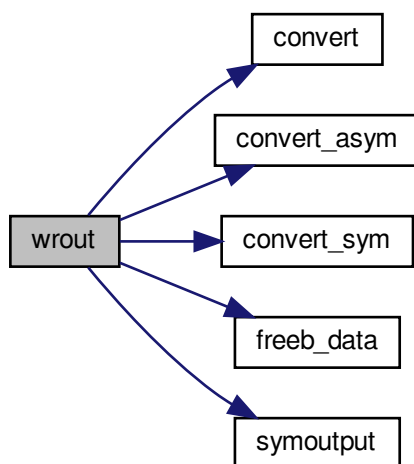
Definition at line 2 of file wrout.f90.

References vmec_io::aminor_p, vforces::armn_e, vforces::armn_o, vmec_main::aspect, vforces::azmn_e, vmec_↵
io::b0, vmec_main::bdotgradv, vmec_main::beta_vol, vmec_io::betapol, vmec_io::betator, vmec_io::betatot, vmec_↵
_io::betaxis, vforces::brmn_e, vmec_main::buc0, vmec_main::bvco, vforces::bzmn_e, vparams::c2p0, vmec_↵
_main::chi, vmec_main::chipf, convert(), convert_asym(), convert_sym(), vmec_persistent::cosmui, vmec_↵
_persistent::cosnv, vparams::cp5, vmec_main::ctor, mgrid_mod::curlabel, vforces::czmn_o, vmecier::dcurr,
vmecier::dgeod, vmecier::dmerc, vmecier::dshear, vmecier::dwell, vmec_main::equif, freeb_data(), vmec_↵
_main::fsql, vmec_main::fsqr, vmec_main::fsqz, vmec_input::ftol_array, vmec_io::ionlarmor, vmec_main::iotaf,
vmec_main::iotas, vmec_main::irzloff, vmec_main::iter2, vmec_main::itfsq, vmec_main::jcuru, vmec_main_↵
::jcurv, vmec_main::jdotb, vmec_params::lamscale, read_wout_mod::ln_abeta, read_wout_mod::ln_ac, read_↵
wout_mod::ln_ac_aux_f, read_wout_mod::ln_ac_aux_s, read_wout_mod::ln_actit, read_wout_mod::ln_ai, read_↵
_wout_mod::ln_ai_aux_f, read_wout_mod::ln_ai_aux_s, read_wout_mod::ln_am, read_wout_mod::ln_am_aux_f,
read_wout_mod::ln_am_aux_s, read_wout_mod::ln_amin, read_wout_mod::ln_aspect, read_wout_mod::ln_asym,
read_wout_mod::ln_b0, read_wout_mod::ln_beta, read_wout_mod::ln_betah, read_wout_mod::ln_bgrv, read_↵
wout_mod::ln_bmnc, read_wout_mod::ln_bmns, read_wout_mod::ln_bsubsmnc, read_wout_mod::ln_bsubsmns,
read_wout_mod::ln_bsubumnc, read_wout_mod::ln_bsubumns, read_wout_mod::ln_bsubvmnc, read_wout_mod_↵
::ln_bsubvmns, read_wout_mod::ln_bsupumnc, read_wout_mod::ln_bsupumns, read_wout_mod::ln_bsupvmnc,
read_wout_mod::ln_bsupvmns, read_wout_mod::ln_buco, read_wout_mod::ln_bvco, read_wout_mod::ln_chi,
read_wout_mod::ln_chipf, read_wout_mod::ln_ctor, read_wout_mod::ln_curlab, read_wout_mod::ln_equif, read_↵
_wout_mod::ln_error, read_wout_mod::ln_extcur, read_wout_mod::ln_extension, read_wout_mod::ln_fp, read_↵
wout_mod::ln_free, read_wout_mod::ln_fsqr, read_wout_mod::ln_gam, read_wout_mod::ln_gmnc, read_wout_↵
mod::ln_gmns, read_wout_mod::ln_iotaf, read_wout_mod::ln_iotah, read_wout_mod::ln_jcuru, read_wout_mod_↵
::ln_jcurv, read_wout_mod::ln_jdotb, read_wout_mod::ln_lar, read_wout_mod::ln_lmnc, read_wout_mod::ln_lmns,
read_wout_mod::ln_magen, read_wout_mod::ln_mass, read_wout_mod::ln_maxit, read_wout_mod::ln_maxmod,
read_wout_mod::ln_maxmod_nyq, read_wout_mod::ln_maxr, read_wout_mod::ln_maxz, read_wout_mod::ln_↵
mcurr, read_wout_mod::ln_merc, read_wout_mod::ln_mgeo, read_wout_mod::ln_mgrid, read_wout_mod::ln_minr,
read_wout_mod::ln_modb, read_wout_mod::ln_mse, read_wout_mod::ln_mshear, read_wout_mod::ln_mwell,
read_wout_mod::ln_pbeta, read_wout_mod::ln_pcurr_type, read_wout_mod::ln_phi, read_wout_mod::ln_phip,

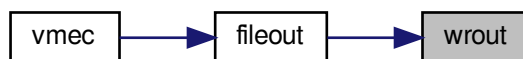
read_wout_mod::ln_phipf, read_wout_mod::ln_piota_type, read_wout_mod::ln_pmass_type, read_wout_mod::ln_←
 _pmod, read_wout_mod::ln_pmod_nyq, read_wout_mod::ln_polmod, read_wout_mod::ln_potvac, read_wout_←
 mod::ln_presf, read_wout_mod::ln_presh, read_wout_mod::ln_qfact, read_wout_mod::ln_racc, read_wout_mod_←
 ::ln_racs, read_wout_mod::ln_radnod, read_wout_mod::ln_rbc, read_wout_mod::ln_rbs, read_wout_mod::ln_rbt0,←
 read_wout_mod::ln_rbt1, read_wout_mod::ln_rmaj, read_wout_mod::ln_rmnc, read_wout_mod::ln_rmns, read_←
 wout_mod::ln_sgs, read_wout_mod::ln_specw, read_wout_mod::ln_tbeta, read_wout_mod::ln_therm, read_wout_←
 _mod::ln_thom, read_wout_mod::ln_tmod, read_wout_mod::ln_tmod_nyq, read_wout_mod::ln_tormod, read_←
 wout_mod::ln_version, read_wout_mod::ln_vol, read_wout_mod::ln_vp, read_wout_mod::ln_wdot, read_wout_←
 _mod::ln_zacc, read_wout_mod::ln_zacs, read_wout_mod::ln_zbc, read_wout_mod::ln_zbs, read_wout_mod_←
 ::ln_zmnc, read_wout_mod::ln_zmns, vmec_main::lthreed, vmec_main::mass, mgrid_mod::mgrid_mode, vmec_←
 _params::mnyq, vmec_params::mscale, vmec_main::neqs, mgrid_mod::nextcur, vmec_params::nnyq, vmec_←
 input::ns_array, vmec_params::nscale, vmec_params::ntmax, vmec_main::overr, vmec_main::phi, realspace::phip,←
 vmec_main::phipf, vmec_main::phips, vacmod::potvac, vmec_main::pres, vmec_main::presf, vmec_main::r0scale,←
 vmec_main::rbtor, vmec_main::rbtor0, vmec_io::rmajor_p, vmec_io::rmax_surf, vmec_io::rmin_surf, vmec_←
 params::rsc, vmec_params::rss, vmec_params::signgs, vmec_persistent::sinmui, vmec_persistent::sinnv, vmec_←
 main::sm, vmec_main::sp, vmec_main::specw, symoutput(), vmec_params::version, read_wout_mod::vn_abeta,←
 read_wout_mod::vn_ac, read_wout_mod::vn_ac_aux_f, read_wout_mod::vn_ac_aux_s, read_wout_mod::vn_actit,←
 read_wout_mod::vn_ai, read_wout_mod::vn_ai_aux_f, read_wout_mod::vn_ai_aux_s, read_wout_mod::vn_am,←
 read_wout_mod::vn_am_aux_f, read_wout_mod::vn_am_aux_s, read_wout_mod::vn_amin, read_wout_mod_←
 ::vn_aspect, read_wout_mod::vn_asym, read_wout_mod::vn_b0, read_wout_mod::vn_beta, read_wout_mod_←
 ::vn_betah, read_wout_mod::vn_bgrv, read_wout_mod::vn_bmnc, read_wout_mod::vn_bmns, read_wout_mod_←
 ::vn_bsubsmnc, read_wout_mod::vn_bsubsmns, read_wout_mod::vn_bsubumnc, read_wout_mod::vn_bsubumns,←
 read_wout_mod::vn_bsubvmnc, read_wout_mod::vn_bsubvmns, read_wout_mod::vn_bsupumnc, read_wout_←
 mod::vn_bsupumns, read_wout_mod::vn_bsupvmnc, read_wout_mod::vn_bsupvmns, read_wout_mod::vn_buco,←
 read_wout_mod::vn_bvco, read_wout_mod::vn_chi, read_wout_mod::vn_chipf, read_wout_mod::vn_ctor, read_←
 wout_mod::vn_curlab, read_wout_mod::vn_equif, read_wout_mod::vn_error, read_wout_mod::vn_extcur, read_←
 wout_mod::vn_extension, read_wout_mod::vn_fp, read_wout_mod::vn_free, read_wout_mod::vn_fsq, read_wout_←
 _mod::vn_fsqr, read_wout_mod::vn_fsqr, read_wout_mod::vn_fsqz, read_wout_mod::vn_ftolv, read_wout_mod_←
 ::vn_gam, read_wout_mod::vn_gmnc, read_wout_mod::vn_gmns, read_wout_mod::vn_iotaf, read_wout_mod_←
 ::vn_iotah, read_wout_mod::vn_jcuru, read_wout_mod::vn_jcurv, read_wout_mod::vn_jdotb, read_wout_mod::vn_←
 _lar, read_wout_mod::vn_lmnc, read_wout_mod::vn_lmns, read_wout_mod::vn_magen, read_wout_mod::vn_←
 mass, read_wout_mod::vn_maxit, read_wout_mod::vn_maxmod, read_wout_mod::vn_maxmod_nyq, read_wout_←
 _mod::vn_maxr, read_wout_mod::vn_maxz, read_wout_mod::vn_mcurr, read_wout_mod::vn_merc, read_wout_←
 _mod::vn_mgeo, mgrid_mod::vn_mgmode, read_wout_mod::vn_mgrid, read_wout_mod::vn_minr, read_wout_←
 mod::vn_modb, read_wout_mod::vn_mshear, read_wout_mod::vn_mwell, mgrid_mod::vn_nextcur, read_wout_←
 _mod::vn_overr, read_wout_mod::vn_pbeta, read_wout_mod::vn_pcurr_type, read_wout_mod::vn_phi, read_←
 wout_mod::vn_phip, read_wout_mod::vn_phipf, read_wout_mod::vn_piota_type, read_wout_mod::vn_pmass_type,←
 read_wout_mod::vn_pmod, read_wout_mod::vn_pmod_nyq, read_wout_mod::vn_polmod, read_wout_mod::vn_←
 _potvac, read_wout_mod::vn_presf, read_wout_mod::vn_presh, read_wout_mod::vn_qfact, read_wout_mod::vn_←
 _racc, read_wout_mod::vn_racs, read_wout_mod::vn_radnod, read_wout_mod::vn_rbc, read_wout_mod::vn_←
 rbs, read_wout_mod::vn_rbt0, read_wout_mod::vn_rbt1, read_wout_mod::vn_rmaj, read_wout_mod::vn_rmnc,←
 read_wout_mod::vn_rmns, read_wout_mod::vn_sgs, read_wout_mod::vn_specw, read_wout_mod::vn_tbeta,←
 read_wout_mod::vn_therm, read_wout_mod::vn_tmod, read_wout_mod::vn_tmod_nyq, read_wout_mod::vn_←
 tormod, read_wout_mod::vn_version, read_wout_mod::vn_vol, read_wout_mod::vn_vp, read_wout_mod::vn_←
 wdot, read_wout_mod::vn_zacc, read_wout_mod::vn_zacs, read_wout_mod::vn_zbc, read_wout_mod::vn_zbs,←
 read_wout_mod::vn_zmnc, read_wout_mod::vn_zmns, vmec_io::volavgb, vmec_io::volume_p, vmec_main::vp,←
 vmec_main::wb, vmec_main::wp, xstuff::xc, vmec_persistent::xm, vmec_persistent::xm_nyq, vmec_persistent::xn,←
 vmec_persistent::xn_nyq, realspace::z1, vmec_params::zcc, vmec_params::zcs, and vmec_io::zmax_surf.

Referenced by fileout().

Here is the call graph for this function:



Here is the caller graph for this function:



Index

- abcoil
 - mgrid_mod, [22](#)
- ac
 - read_wout_mod, [81](#)
 - vmec_input, [217](#)
- ac_aux_f
 - read_wout_mod, [81](#)
 - vmec_input, [217](#)
- ac_aux_s
 - read_wout_mod, [81](#)
 - vmec_input, [218](#)
- actemp
 - vacmod, [180](#)
- add_fluxes
 - add_fluxes.f90, [289](#)
- add_fluxes.f90
 - add_fluxes, [289](#)
- add_subdirectory
 - CMakeLists.txt, [301](#), [302](#)
- adm
 - vacmod, [180](#)
- adp
 - vacmod, [180](#)
- ai
 - read_wout_mod, [82](#)
 - vmec_input, [218](#)
- ai_aux_f
 - read_wout_mod, [82](#)
 - vmec_input, [218](#)
- ai_aux_s
 - read_wout_mod, [82](#)
 - vmec_input, [218](#)
- alfa
 - read_wout_mod, [82](#)
- alias
 - alias.f90, [290](#)
- alias.f90
 - alias, [290](#)
- all_slm
 - vacmod, [181](#)
- all_slp
 - vacmod, [181](#)
- all_tlm
 - vacmod, [181](#)
- all_tlp
 - vacmod, [181](#)
- allocate_func3d
 - allocate_func3d.f90, [291](#)
- allocate_func3d.f90
 - allocate_func3d, [291](#)
- allocate_nestor
 - vacmod, [179](#)
- allocate_ns
 - allocate_ns.f90, [293](#)
- allocate_ns.f90
 - allocate_ns, [293](#)
- allocate_nunv
 - allocate_nunv.f90, [294](#)
- allocate_nunv.f90
 - allocate_nunv, [294](#)
- alp
 - vacmod, [181](#)
- alu
 - vacmod, [182](#)
- alv
 - vacmod, [182](#)
- alvp
 - vacmod, [182](#)
- am
 - read_wout_mod, [82](#)
 - vmec_input, [218](#)
- am_aux_f
 - read_wout_mod, [83](#)
 - vmec_input, [219](#)
- am_aux_s
 - read_wout_mod, [83](#)
 - vmec_input, [219](#)
- amatrix
 - vacmod, [182](#)
- amatsav
 - vacmod, [182](#)
- amatsav_nestor
 - nestor_io, [42](#)
- amatsavdim
 - nestor_io, [42](#)
- aminor
 - read_wout_mod, [83](#)
- aminor_p
 - vmec_io, [228](#)
- analysum
 - analysum.f90, [358](#)
- analysum.f90
 - analysum, [358](#)
- analysum2
 - analysum2.f90, [359](#)
- analysum2.f90
 - analysum2, [359](#)
- analyt

analyt.f90, [359](#)
 analyt.f90
 analyt, [359](#)
 anglemse
 read_wout_mod, [83](#)
 aphi
 vmec_input, [219](#)
 ard
 vmec_main, [235](#)
 arm
 vmec_main, [235](#)
 armn
 vforces, [207](#)
 armn_e
 vforces, [207](#)
 armn_o
 vforces, [208](#)
 aspect
 read_wout_mod, [83](#)
 vmec_main, [235](#)
 aspectratio
 aspectratio.f90, [295](#)
 aspectratio.f90
 aspectratio, [295](#)
 assign_bptrs
 mgrid_mod, [19](#)
 astemp
 vacmod, [183](#)
 auu
 vacmod, [183](#)
 auv
 vacmod, [183](#)
 avv
 vacmod, [183](#)
 azd
 vmec_main, [235](#)
 azm
 vmec_main, [235](#)
 azm1u
 vacmod, [183](#)
 azmn
 vforces, [208](#)
 azmn_e
 vforces, [208](#)
 azmn_o
 vforces, [208](#)
 azp1u
 vacmod, [184](#)

 b0
 read_wout_mod, [84](#)
 vmec_io, [228](#)
 b_chi
 mgrid_mod, [22](#)
 bad_jacobian_flag
 vmec_params, [261](#)
 bbc
 read_wout_mod, [84](#)
 bcoil
 mgrid_mod, [22](#)
 bcos
 vacmod, [184](#)
 bcovar
 bcovar.f90, [296](#)
 bcovar.f90
 bcovar, [296](#)
 bcwgt
 read_wout_mod, [84](#)
 bdamp
 vmec_main, [235](#)
 bdotb
 vmec_main, [236](#)
 bdotgradv
 read_wout_mod, [84](#)
 vmec_main, [236](#)
 becoil
 becoil.f90, [361](#)
 becoil.f90
 becoil, [361](#)
 belicu
 belicu.f90, [361](#)
 belicu.f90
 belicu, [361](#)
 beta_vol
 read_wout_mod, [84](#)
 vmec_main, [236](#)
 betapol
 read_wout_mod, [85](#)
 vmec_io, [229](#)
 betator
 read_wout_mod, [85](#)
 vmec_io, [229](#)
 betatot
 read_wout_mod, [85](#)
 vmec_io, [229](#)
 betaxis
 read_wout_mod, [85](#)
 vmec_io, [229](#)
 bexn
 vacmod, [184](#)
 bexni
 vacmod, [184](#)
 bextern
 bextern.f90, [362](#)
 bextern.f90
 bextern, [362](#)
 bextrema
 bextrema.f90, [298](#)
 bextrema.f90
 bextrema, [298](#)
 bexu
 vacmod, [184](#)
 bexv
 vacmod, [185](#)
 blam
 vmec_main, [236](#)
 blmn

vforces, 208
blmn_e
 vforces, 209
blmn_o
 vforces, 209
bloat
 vmec_input, 219
bloopnames
 mgrid_mod, 22
bmax
 vmec_main, 236
bmin
 vmec_main, 237
bmnc
 read_wout_mod, 85
bmns
 read_wout_mod, 86
bpedge
 vmec_main, 237
bphi
 vacmod, 185
bphiv
 vacmod, 185
bpvac
 mgrid_mod, 22
brad
 vacmod, 185
brd
 vmec_main, 237
bredg
 vmec_main, 237
brm
 vmec_main, 237
brmn
 vforces, 209
brmn_e
 vforces, 209
brmn_o
 vforces, 209
brv
 vacmod, 185
brvac
 mgrid_mod, 22
bsin
 vacmod, 186
bsqsav
 vmec_main, 238
bsqvac
 vacmod, 186
bss
 bss.f90, 299
bss.f90
 bss, 299
bsubsmnc
 read_wout_mod, 86
bsubsmns
 read_wout_mod, 86
bsubu
 vacmod, 186
bsubu0
 vmec_main, 238
bsubu_sur
 vac_persistent, 171
bsubumnc
 read_wout_mod, 86
bsubumns
 read_wout_mod, 86
bsubv
 vacmod, 186
bsubv_sur
 vac_persistent, 171
bsubvmnc
 read_wout_mod, 87
bsubvmns
 read_wout_mod, 87
bsubvvac
 vacmod, 186
bsubvvac_nestor
 nestor_io, 42
bsupu_sur
 vac_persistent, 171
bsupumnc
 read_wout_mod, 87
bsupumns
 read_wout_mod, 87
bsupv_sur
 vac_persistent, 172
bsupvmnc
 read_wout_mod, 87
bsupvmns
 read_wout_mod, 88
buc
 read_wout_mod, 88
 vmec_main, 238
bucof
 vmec_main, 238
bvac
 mgrid_mod, 23
bvco
 read_wout_mod, 88
 vmec_main, 238
bvcof
 vmec_main, 239
bvecsav
 vacmod, 187
bvecsav_nestor
 nestor_io, 43
bvecsavdim
 nestor_io, 43
bz
 vacmod, 187
bzd
 vmec_main, 239
bzedg
 vmec_main, 239
bzm

- vmec_main, 239
- bzmn
 - vforces, 210
- bzmn_e
 - vforces, 210
- bzmn_o
 - vforces, 210
- bzv
 - vacmod, 187
- bzvac
 - mgrid_mod, 23
- c1pm13
 - vparams, 277
- c1pm2
 - vparams, 277
- c1pm8
 - vparams, 277
- c2p0
 - vparams, 277
- c3p0
 - vparams, 277
- calc_fbal
 - calc_fbal.f90, 300
- calc_fbal.f90
 - calc_fbal, 300
- cbig
 - vparams, 277
- chi
 - read_wout_mod, 88
 - vmec_main, 239
- chip
 - realspace, 162
- chipf
 - read_wout_mod, 88
 - vmec_main, 240
- chips
 - vmec_main, 240
- circum_p
 - vmec_io, 229
- clam
 - vmec_main, 240
- cleanup_flag
 - vmec_params, 261
- clmn
 - vforces, 210
- clmn_e
 - vforces, 210
- clmn_o
 - vforces, 211
- cma
 - vacmod, 187
- cma11u
 - vacmod, 187
- CMakeLists.txt
 - add_subdirectory, 301, 302
 - list, 302
- cmns
 - vac_persistent, 172
- compute_currents
 - read_wout_mod, 79
- condif
 - mgrid_mod, 23
- constrain_m1
 - residue.f90, 402
- convert
 - convert.f90, 303
- convert.f90
 - convert, 303
- convert_asym
 - totzsp.f90, 420
- convert_sym
 - totzsp.f90, 420
- cos01
 - vmec_persistent, 270
- cosmu
 - vmec_persistent, 271
- cosmui
 - vmec_persistent, 271
- cosmui3
 - vmec_persistent, 271
- cosmum
 - vmec_persistent, 271
- cosmumi
 - vmec_persistent, 271
- cosmumi3
 - vmec_persistent, 271
- cosnv
 - vmec_persistent, 272
- cosnvn
 - vmec_persistent, 272
- cosper
 - vac_persistent, 172
- cosu
 - vac_persistent, 172
- cosu1
 - vac_persistent, 172
- cosui
 - vac_persistent, 173
- cosuv
 - vac_persistent, 173
- cosv
 - vac_persistent, 173
- cosv1
 - vac_persistent, 173
- cp05
 - vparams, 278
- cp15
 - vparams, 278
- cp25
 - vparams, 278
- cp5
 - vparams, 278
- cprec
 - stel_kinds, 170
- crd
 - vmec_main, 240

crmn
 vforces, 211
 crmn_e
 vforces, 211
 crmn_o
 vforces, 211
 cross_area_p
 vmec_io, 230
 csign
 vac_persistent, 173
 ctor
 nestor_io, 43
 vmec_main, 240
 curlabel
 mgrid_mod, 23
 curmid
 read_wout_mod, 89
 currmnc
 read_wout_mod, 89
 currmns
 read_wout_mod, 89
 currv
 vmec_main, 241
 currmnc
 read_wout_mod, 89
 currmns
 read_wout_mod, 89
 curtor
 vmec_input, 219
 czmn
 vforces, 211
 czmn_e
 vforces, 212
 czmn_o
 vforces, 212

 datastark
 read_wout_mod, 90
 datathom
 read_wout_mod, 90
 dbcoil
 mgrid_mod, 23
 dbsq
 vmec_main, 241
 dcurr
 read_wout_mod, 90
 vmecier, 275
 degree
 stel_constants, 168
 delbsq
 vmec_main, 241
 delphid
 read_wout_mod, 90
 delrb
 mgrid_mod, 24
 delt
 vmec_input, 220
 delt0r
 vmec_main, 241

 delt1u
 vacmod, 188
 delzb
 mgrid_mod, 24
 dgeod
 read_wout_mod, 90
 vmecier, 275
 dlam
 vmec_main, 241
 dmerc
 read_wout_mod, 90
 vmecier, 275
 dp
 stel_kinds, 170
 drv
 vacmod, 188
 dsave
 vacmod, 188
 dshear
 read_wout_mod, 91
 vmecier, 275
 dsiext
 mgrid_mod, 24
 dsilabel
 mgrid_mod, 24
 dsiobt
 read_wout_mod, 91
 dwell
 read_wout_mod, 91
 vmecier, 276

 elongation
 elongation.f90, 316
 elongation.f90
 elongation, 316
 eqfor
 eqfor.f90, 317
 eqfor.f90
 eqfor, 317
 eqsolve
 eqsolve.f90, 319
 eqsolve.f90
 eqsolve, 319
 equip
 read_wout_mod, 91
 vmec_main, 242
 evolve
 evolve.f90, 321
 evolve.f90
 evolve, 321
 extcur
 nestor_io, 43
 read_wout_mod, 91
 vmec_input, 220
 extra1
 realspace, 162
 extra2
 realspace, 162
 extra3

- realspace, 162
- extra4
 - realspace, 162
- faccon
 - vmec_main, 242
- faclam
 - vmec_main, 242
- faclam0
 - vmec_main, 242
- fbal, 11
 - frcc_fac, 11
 - fzsc_fac, 11
 - rru_fac, 11
 - rzu_fac, 12
- fedge
 - vmec_main, 242
- fext_fft
 - fsym_fft.f90, 332
- fileout
 - fileout.f90, 323
- fileout.f90
 - fileout, 323
- fixaray
 - fixaray.f90, 325
- fixaray.f90
 - fixaray, 325
- flip_theta
 - flip_theta.f90, 326
- flip_theta.f90
 - flip_theta, 326
- flmwgt
 - read_wout_mod, 92
- fnorm
 - vmec_main, 243
- fnorm1
 - vmec_main, 243
- fnorml
 - vmec_main, 243
- forces
 - forces.f90, 327
- forces.f90
 - forces, 327
- fourl
 - fourl.f90, 372
- fourl.f90
 - fourl, 372
- fourp
 - fourp.f90, 373
- fourp.f90
 - fourp, 373
- frcc_fac
 - fbal, 11
- free_mem_funct3d
 - free_mem_funct3d.f90, 328
- free_mem_funct3d.f90
 - free_mem_funct3d, 328
- free_mem_nestor
 - vacmod, 179
- free_mem_ns
 - free_mem_ns.f90, 329
- free_mem_ns.f90
 - free_mem_ns, 329
- free_mem_nunv
 - free_mem_nunv.f90, 330
- free_mem_nunv.f90
 - free_mem_nunv, 330
- free_mgrid
 - mgrid_mod, 20
- freeb_data
 - freeb_data.f90, 331
- freeb_data.f90
 - freeb_data, 331
- fsq
 - vmec_main, 243
- fsql
 - read_wout_mod, 92
 - vmec_main, 243
- fsql1
 - vmec_main, 244
- fsqr
 - read_wout_mod, 92
 - vmec_main, 244
- fsqr1
 - vmec_main, 244
- fsqsum0
 - vmec_main, 244
- fsqt
 - read_wout_mod, 92
- fsqz
 - read_wout_mod, 92
 - vmec_main, 244
- fsqz1
 - vmec_main, 245
- fsym_fft
 - fsym_fft.f90, 333
- fsym_fft.f90
 - fext_fft, 332
 - fsym_fft, 333
- fsym_invfft
 - fsym_invfft.f90, 334
- fsym_invfft.f90
 - fsym_invfft, 334
- ftol_array
 - vmec_input, 220
- ftol_default
 - vmec_input, 220
- ftolv
 - read_wout_mod, 93
 - vmec_main, 245
- funct3d
 - funct3d.f90, 335
- funct3d.f90
 - funct3d, 335
- function_test
 - functions, 12
- functions, 12

- function_test, 12
 - two_power, 13
 - two_power_gs, 13
- fzsc_fac
 - fbal, 11
- g1
 - vacmod, 188
- g2
 - vacmod, 188
- ga1
 - vacmod, 189
- ga2
 - vacmod, 189
- gamma
 - read_wout_mod, 93
 - vmec_input, 220
- gc
 - xstuff, 282
- gcon
 - realspace, 163
- getbsubs
 - getbsubs.f90, 338
- getbsubs.f90
 - getbsubs, 338
- getcurmid
 - getcurmid.f90, 339
- getcurmid.f90
 - getcurmid, 339
- getfsq
 - getfsq.f90, 340
- getfsq.f90
 - getfsq, 340
- gmnc
 - read_wout_mod, 93
- gmns
 - read_wout_mod, 93
- green
 - vacmod, 189
- greenf
 - greenf.f90, 373
- greenf.f90
 - greenf, 373
- greenp
 - vacmod, 189
- grpmn
 - vacmod, 189
- grpmn_m_map_wrt
 - vacmod, 190
- grpmn_n_map_wrt
 - vacmod, 190
- gsave
 - vacmod, 190
- gstore
 - vacmod, 190
- guess_axis
 - guess_axis.f90, 341
- guess_axis.f90
 - guess_axis, 341
- guu
 - realspace, 163
- guu_b
 - vacmod, 190
- guv
 - realspace, 163
- guv_b
 - vacmod, 191
- gvv
 - realspace, 163
- gvv_b
 - vacmod, 191
- heading
 - heading.f90, 342
- heading.f90
 - heading, 342
- hi
 - vmec_main, 245
- hp
 - vmec_main, 245
- hs
 - vmec_main, 245
- iasym
 - read_wout_mod, 93
- iconnect
 - mgrid_mod, 24
- icurv
 - vmec_main, 246
- iequi
 - vmec_main, 246
- ier_flag
 - nestor_io, 43
- ierr_vmec
 - read_wout_mod, 94
- ijacob
 - vmec_main, 246
- imatch_phiedge
 - read_wout_mod, 94
- imirr
 - vac_persistent, 174
- imse
 - read_wout_mod, 94
- indata0
 - vparams, 278
- initialize_radial
 - initialize_radial.f90, 343
- initialize_radial.f90
 - initialize_radial, 343
- input_error_flag
 - vmec_params, 262
- input_extension
 - nestor_io, 44
 - read_wout_mod, 94
 - vmec_input, 221
- interp
 - interp.f90, 345
- interp.f90

- interp, 345
- ionlarmor
 - read_wout_mod, 94
 - vmec_io, 230
- iotaf
 - read_wout_mod, 95
 - vmec_main, 246
- iotas
 - read_wout_mod, 95
 - vmec_main, 247
- ipnodes
 - read_wout_mod, 95
- iprec
 - stel_kinds, 170
- ireflect
 - vmec_main, 247
- irst
 - vmec_main, 247
- irzloff
 - vmec_main, 247
- isigng
 - read_wout_mod, 95
- isnodes
 - read_wout_mod, 95
- iter1
 - vmec_main, 248
- iter2
 - vmec_main, 248
- itfsq
 - read_wout_mod, 96
 - vmec_main, 248
- itor
 - read_wout_mod, 96
- itse
 - read_wout_mod, 96
- ivac
 - nestor_io, 44
 - vmec_main, 248
- ivacskip
 - nestor_io, 44
- ixm
 - vmec_persistent, 272
- jac75_flag
 - vmec_params, 262
- jacobian
 - jacobian.f90, 346
- jacobian.f90
 - jacobian, 346
- jcuru
 - read_wout_mod, 96
 - vmec_main, 249
- jcurv
 - read_wout_mod, 96
 - vmec_main, 249
- jdotb
 - read_wout_mod, 97
 - vmec_main, 249
- jlam
 - vmec_params, 262
- jmin1
 - vmec_params, 262
- jmin2
 - vmec_params, 262
- jmin3
 - vmec_persistent, 272
- jpar2
 - vmec_main, 249
- jperp2
 - vmec_main, 249
- jxbforce
 - jxbforce.f90, 347
- jxbforce.f90
 - jxbforce, 347
- jxbout0
 - vparams, 278
- kappa_p
 - vmec_io, 230
- lamcal
 - lamcal.f90, 349
- lamcal.f90
 - lamcal, 349
- lamscale
 - vmec_params, 263
- lasym
 - nestor_io, 44
 - read_wout_mod, 97
 - vmec_input, 221
- lbsubs
 - vmec_input, 221
- lconm1
 - vmec_main, 250
- lflip
 - vmec_main, 250
- lfreeb
 - vmec_input, 221
- limitr
 - mgrid_mod, 25
- line_seg
 - line_segment, 16
- line_seg_int
 - line_segment, 16
- line_seg_test
 - line_segment, 17
- line_segment
 - line_seg, 15
 - line_seg, 16
 - line_seg_int, 16
 - line_seg_test, 17
- list
 - CMakeLists.txt, 302
- lmnc
 - read_wout_mod, 97
- lmns
 - read_wout_mod, 97
- ln_abeta
 - read_wout_mod, 97

In_ac
 read_wout_mod, 98
In_ac_aux_f
 read_wout_mod, 98
In_ac_aux_s
 read_wout_mod, 98
In_actit
 read_wout_mod, 98
In_ai
 read_wout_mod, 98
In_ai_aux_f
 read_wout_mod, 99
In_ai_aux_s
 read_wout_mod, 99
In_am
 read_wout_mod, 99
In_am_aux_f
 read_wout_mod, 99
In_am_aux_s
 read_wout_mod, 99
In_amin
 read_wout_mod, 100
In_aspect
 read_wout_mod, 100
In_asym
 read_wout_mod, 100
In_b0
 read_wout_mod, 100
In_beta
 read_wout_mod, 100
In_betah
 read_wout_mod, 101
In_bgrv
 read_wout_mod, 101
In_bmnc
 read_wout_mod, 101
In_bmns
 read_wout_mod, 101
In_bsubsmnc
 read_wout_mod, 101
In_bsubsmns
 read_wout_mod, 102
In_bsubumnc
 read_wout_mod, 102
In_bsubumnc_sur
 read_wout_mod, 102
In_bsubumns
 read_wout_mod, 102
In_bsubumns_sur
 read_wout_mod, 102
In_bsubvmnc
 read_wout_mod, 103
In_bsubvmnc_sur
 read_wout_mod, 103
In_bsubvmns
 read_wout_mod, 103
In_bsubvmns_sur
 read_wout_mod, 103
In_bsupumnc
 read_wout_mod, 103
In_bsupumnc_sur
 read_wout_mod, 104
In_bsupumns
 read_wout_mod, 104
In_bsupumns_sur
 read_wout_mod, 104
In_bsupvmnc
 read_wout_mod, 104
In_bsupvmnc_sur
 read_wout_mod, 104
In_bsupvmns
 read_wout_mod, 105
In_bsupvmns_sur
 read_wout_mod, 105
In_buco
 read_wout_mod, 105
In_bvco
 read_wout_mod, 105
In_chi
 read_wout_mod, 105
In_chipf
 read_wout_mod, 106
In_ctor
 read_wout_mod, 106
In_curlab
 read_wout_mod, 106
In_equip
 read_wout_mod, 106
In_error
 read_wout_mod, 106
In_extcur
 read_wout_mod, 107
In_extension
 read_wout_mod, 107
In_fp
 read_wout_mod, 107
In_free
 read_wout_mod, 107
In_fsq
 read_wout_mod, 107
In_fsql
 read_wout_mod, 108
In_fsqr
 read_wout_mod, 108
In_fsqz
 read_wout_mod, 108
In_ftolv
 read_wout_mod, 108
In_gam
 read_wout_mod, 108
In_gmnc
 read_wout_mod, 108
In_gmns
 read_wout_mod, 109
In_iotaf
 read_wout_mod, 109

- In_iotah
 - read_wout_mod, 109
- In_jcuru
 - read_wout_mod, 109
- In_jcurv
 - read_wout_mod, 109
- In_jdotb
 - read_wout_mod, 110
- In_lar
 - read_wout_mod, 110
- In_lmnc
 - read_wout_mod, 110
- In_lmns
 - read_wout_mod, 110
- In_magen
 - read_wout_mod, 110
- In_mass
 - read_wout_mod, 111
- In_maxit
 - read_wout_mod, 111
- In_maxmod
 - read_wout_mod, 111
- In_maxmod_nyq
 - read_wout_mod, 111
- In_maxr
 - read_wout_mod, 111
- In_maxz
 - read_wout_mod, 112
- In_mcurr
 - read_wout_mod, 112
- In_merc
 - read_wout_mod, 112
- In_mgeo
 - read_wout_mod, 112
- In_mgrid
 - read_wout_mod, 112
- In_minr
 - read_wout_mod, 113
- In_modb
 - read_wout_mod, 113
- In_mse
 - read_wout_mod, 113
- In_mshear
 - read_wout_mod, 113
- In_mwell
 - read_wout_mod, 113
- In_next
 - mgrid_mod, 25
- In_pbeta
 - read_wout_mod, 114
- In_pcurr_type
 - read_wout_mod, 114
- In_phi
 - read_wout_mod, 114
- In_phip
 - read_wout_mod, 114
- In_phipf
 - read_wout_mod, 114
- In_piota_type
 - read_wout_mod, 115
- In_pmass_type
 - read_wout_mod, 115
- In_pmod
 - read_wout_mod, 115
- In_pmod_nyq
 - read_wout_mod, 115
- In_polmod
 - read_wout_mod, 115
- In_potvac
 - read_wout_mod, 116
- In_presf
 - read_wout_mod, 116
- In_presh
 - read_wout_mod, 116
- In_qfact
 - read_wout_mod, 116
- In_racc
 - read_wout_mod, 116
- In_racs
 - read_wout_mod, 117
- In_radnod
 - read_wout_mod, 117
- In_rbc
 - read_wout_mod, 117
- In_rbs
 - read_wout_mod, 117
- In_rbt0
 - read_wout_mod, 117
- In_rbt1
 - read_wout_mod, 118
- In_recon
 - read_wout_mod, 118
- In_rmaj
 - read_wout_mod, 118
- In_rmnc
 - read_wout_mod, 118
- In_rmns
 - read_wout_mod, 118
- In_sgs
 - read_wout_mod, 119
- In_specw
 - read_wout_mod, 119
- In_tbeta
 - read_wout_mod, 119
- In_therm
 - read_wout_mod, 119
- In_thom
 - read_wout_mod, 119
- In_tmod
 - read_wout_mod, 120
- In_tmod_nyq
 - read_wout_mod, 120
- In_tormod
 - read_wout_mod, 120
- In_version
 - read_wout_mod, 120

ln_vol
 read_wout_mod, 120
 ln_vp
 read_wout_mod, 121
 ln_wdot
 read_wout_mod, 121
 ln_zacc
 read_wout_mod, 121
 ln_zacs
 read_wout_mod, 121
 ln_zbc
 read_wout_mod, 121
 ln_zbs
 read_wout_mod, 122
 ln_zmnc
 read_wout_mod, 122
 ln_zmns
 read_wout_mod, 122
 loadrzi
 read_wout_mod, 80
 lthreed
 read_wout_mod, 122
 vmec_main, 250
 lwout_opened
 read_wout_mod, 122

 m0
 vmec_params, 263
 m1
 vmec_params, 263
 m_map_wrt
 vacmod, 191
 machsq
 read_wout_mod, 123
 magnetic_fluxes.f90
 polflux, 350
 polflux_deriv, 351
 torflux, 352
 torflux_deriv, 353
 mass
 read_wout_mod, 123
 vmec_main, 250
 mercier
 mercier.f90, 354
 mercier.f90
 mercier, 354
 meven
 vmec_params, 263
 mf
 vacmod0, 203
 mf1
 vacmod0, 203
 mfilter_fbdy
 vmec_input, 221
 mgrid_file
 nestor_io, 44
 read_wout_mod, 123
 vmec_input, 222
 mgrid_mod, 17

 abcoil, 22
 assign_bptrs, 19
 b_chi, 22
 bcoil, 22
 bloopnames, 22
 bpvac, 22
 brvac, 22
 bvac, 23
 bzvac, 23
 condif, 23
 curlabel, 23
 dbcoil, 23
 delrb, 24
 delzb, 24
 dsiext, 24
 dsilabel, 24
 free_mgrid, 20
 iconnect, 24
 limitr, 25
 ln_next, 25
 mgrid_mode, 25
 mgrid_path, 25
 mgrid_path_old, 25
 nbcoil_max, 26
 nbcoils, 26
 nbcoilsn, 26
 nbfldn, 26
 nbsets, 26
 nbvac, 26
 needbfld, 27
 needflx, 27
 nextcur, 27
 nfper0, 27
 nlim, 27
 nlim_max, 28
 nlimset, 28
 nobd, 28
 nobser, 28
 np0b, 28
 nr0b, 28
 nrgrid, 29
 nsets, 29
 nsetsn, 29
 nz0b, 29
 nzgrid, 29
 pfcspec, 29
 plbfld, 30
 plflux, 30
 psiext, 30
 raw_coil_current, 30
 rbcoil, 30
 rbcoilsqr, 31
 read_mgrid, 20
 reslim, 31
 rlim, 31
 rmaxb, 31
 rminb, 31
 rx1, 32

- rx2, [32](#)
- seplim, [32](#)
- sum_bfield, [21](#)
- tokid, [32](#)
- unpsiext, [32](#)
- vn_bp0, [32](#)
- vn_br0, [33](#)
- vn_bz0, [33](#)
- vn_coilcur, [33](#)
- vn_coilgrp, [33](#)
- vn_ir, [33](#)
- vn_jz, [34](#)
- vn_kp, [34](#)
- vn_mgmode, [34](#)
- vn_nextcur, [34](#)
- vn_nfp, [34](#)
- vn_rmax, [35](#)
- vn_rmin, [35](#)
- vn_zmax, [35](#)
- vn_zmin, [35](#)
- xobser, [35](#)
- xobsqr, [36](#)
- zbcil, [36](#)
- zlim, [36](#)
- zmaxb, [36](#)
- zminb, [36](#)
- zobser, [37](#)
- zy1, [37](#)
- zy2, [37](#)
- mgrid_mode
 - mgrid_mod, [25](#)
- mgrid_path
 - mgrid_mod, [25](#)
- mgrid_path_old
 - mgrid_mod, [25](#)
- misc_error_flag
 - vmec_params, [264](#)
- mn1dim
 - nestor_io, [45](#)
- mnmax
 - nestor_io, [45](#)
 - read_wout_mod, [123](#)
 - vmec_dim, [212](#)
- mnmax_nyq
 - read_wout_mod, [123](#)
 - vmec_dim, [213](#)
- mnpd
 - vacmod0, [204](#)
- mnpd2
 - vacmod0, [204](#)
- mnpd2_nestor
 - nestor_io, [45](#)
- mnpotdim
 - nestor_io, [45](#)
- mns
 - vmec_dim, [213](#)
- mnsiz
 - vmec_dim, [213](#)
- mnyq
 - read_wout_mod, [124](#)
 - vmec_params, [264](#)
- modd
 - vmec_params, [264](#)
- mpol
 - nestor_io, [45](#)
 - read_wout_mod, [124](#)
 - vmec_input, [222](#)
- mpol1
 - vmec_dim, [213](#)
- mpol1d
 - vparams, [279](#)
- mpol_default
 - vmec_input, [222](#)
- mpold
 - vparams, [279](#)
- mscale
 - vmec_params, [264](#)
- msewgt
 - read_wout_mod, [124](#)
- mu0
 - stel_constants, [168](#)
- multi_ns_grid
 - vmec_main, [250](#)
- n0
 - vmec_params, [264](#)
- n_map_wrt
 - vacmod, [191](#)
- nbcoil_max
 - mgrid_mod, [26](#)
- nbcoils
 - mgrid_mod, [26](#)
- nbcoilsn
 - mgrid_mod, [26](#)
- nbfldn
 - mgrid_mod, [26](#)
- nbsets
 - mgrid_mod, [26](#)
- nbvac
 - mgrid_mod, [26](#)
- ncurr
 - vmec_input, [222](#)
- ndamp
 - vmec_params, [265](#)
- ndatafmax
 - vparams, [279](#)
- ndatai
 - vmec_main, [251](#)
- ndatap
 - vmec_main, [251](#)
- ndim
 - vacmod0, [204](#)
- needbfld
 - mgrid_mod, [27](#)
- needflx
 - mgrid_mod, [27](#)
- neqs

- vmec_main, [251](#)
- nestor
 - nestor_main.f90, [374](#)
- nestor_io, [37](#)
 - amatsav_nestor, [42](#)
 - amatsavdim, [42](#)
 - bsubvvac_nestor, [42](#)
 - bvecsav_nestor, [43](#)
 - bvecsavdim, [43](#)
 - ctor, [43](#)
 - extcur, [43](#)
 - ier_flag, [43](#)
 - input_extension, [44](#)
 - ivac, [44](#)
 - ivacskip, [44](#)
 - lasym, [44](#)
 - mgrid_file, [44](#)
 - mn1dim, [45](#)
 - mnmax, [45](#)
 - mnpd2_nestor, [45](#)
 - mnpotdim, [45](#)
 - mpol, [45](#)
 - nextcur, [46](#)
 - nextcurim, [46](#)
 - nfp, [46](#)
 - ntheta, [46](#)
 - ntor, [46](#)
 - nzeta, [47](#)
 - nzetadim, [47](#)
 - nzntdim, [47](#)
 - r2dim, [47](#)
 - raxis, [47](#)
 - rbtor, [48](#)
 - read_nestor_inputs, [40](#)
 - rmnc, [48](#)
 - rmns, [48](#)
 - signgs, [48](#)
 - vacuum_calls, [48](#)
 - vn_adm, [49](#)
 - vn_adp, [49](#)
 - vn_all_slm, [49](#)
 - vn_all_slp, [49](#)
 - vn_all_tlm, [49](#)
 - vn_all_tlp, [50](#)
 - vn_amatrix, [50](#)
 - vn_amatsav, [50](#)
 - vn_auu, [50](#)
 - vn_auv, [50](#)
 - vn_avv, [51](#)
 - vn_azm1u, [51](#)
 - vn_azp1u, [51](#)
 - vn_bexn, [51](#)
 - vn_bexni, [51](#)
 - vn_bexu, [52](#)
 - vn_bexv, [52](#)
 - vn_bphi, [52](#)
 - vn_bphiv, [52](#)
 - vn_brاد, [52](#)
 - vn_brв, [53](#)
 - vn_bsqvac, [53](#)
 - vn_bsubu, [53](#)
 - vn_bsubv, [53](#)
 - vn_bsubvvac, [53](#)
 - vn_bvecsav, [54](#)
 - vn_bz, [54](#)
 - vn_bzv, [54](#)
 - vn_cma, [54](#)
 - vn_cma11u, [54](#)
 - vn_ctor, [55](#)
 - vn_delt1u, [55](#)
 - vn_drv, [55](#)
 - vn_extcur, [55](#)
 - vn_green, [55](#)
 - vn_greenp, [56](#)
 - vn_grpmn, [56](#)
 - vn_grpmn_m_map, [56](#)
 - vn_grpmn_n_map, [56](#)
 - vn_gstore, [56](#)
 - vn_guu_b, [57](#)
 - vn_guv_b, [57](#)
 - vn_gvv_b, [57](#)
 - vn_ier_flag, [57](#)
 - vn_imirr, [57](#)
 - vn_inputext, [58](#)
 - vn_ivac, [58](#)
 - vn_ivacskip, [58](#)
 - vn_lasym, [58](#)
 - vn_m_map, [58](#)
 - vn_mgrid, [59](#)
 - vn_mnmax, [59](#)
 - vn_mnpd, [59](#)
 - vn_mnpd2, [59](#)
 - vn_mpol, [59](#)
 - vn_n_map, [60](#)
 - vn_nfp, [60](#)
 - vn_ntheta, [60](#)
 - vn_ntor, [60](#)
 - vn_nzeta, [60](#)
 - vn_pmod, [61](#)
 - vn_potu, [61](#)
 - vn_potv, [61](#)
 - vn_potvac, [61](#)
 - vn_r0m, [61](#)
 - vn_r0p, [62](#)
 - vn_r1b, [62](#)
 - vn_r1m, [62](#)
 - vn_r1p, [62](#)
 - vn_ra1m, [62](#)
 - vn_ra1p, [63](#)
 - vn_raxis_nestor, [63](#)
 - vn_rbtor, [63](#)
 - vn_rcosuv, [63](#)
 - vn_rmnc, [63](#)
 - vn_rmns, [64](#)
 - vn_rsinuv, [64](#)
 - vn_rub, [64](#)

- vn_ru, [64](#)
- vn_ruv, [64](#)
- vn_rvb, [65](#)
- vn_rvv, [65](#)
- vn_rzb2, [65](#)
- vn_signgs, [65](#)
- vn_snr, [65](#)
- vn_snv, [66](#)
- vn_snz, [66](#)
- vn_sqad1u, [66](#)
- vn_sqad2u, [66](#)
- vn_sqrt, [66](#)
- vn_sqrtc, [67](#)
- vn_tanu, [67](#)
- vn_tanv, [67](#)
- vn_tmod, [67](#)
- vn_vacuum_calls, [67](#)
- vn_wint, [68](#)
- vn_xmpot, [68](#)
- vn_xnpot, [68](#)
- vn_z1b, [68](#)
- vn_zaxis_nestor, [68](#)
- vn_zmnc, [69](#)
- vn_zmns, [69](#)
- vn_zub, [69](#)
- vn_zuu, [69](#)
- vn_zuv, [69](#)
- vn_zvb, [70](#)
- vn_zvv, [70](#)
- wint, [70](#)
- write_nestor_outputs, [41](#)
- xm, [70](#)
- xn, [70](#)
- zaxis, [71](#)
- zmnc, [71](#)
- zmns, [71](#)
- nestor_io.f90
 - read_nestor_outputs, [366](#)
 - write_nestor_inputs, [367](#)
- nestor_main.f90
 - nestor, [374](#)
- nextcur
 - mgrid_mod, [27](#)
 - nestor_io, [46](#)
- nextcurim
 - nestor_io, [46](#)
- nf
 - vacmod0, [204](#)
- nf1
 - vacmod0, [204](#)
- nfilter_fbdy
 - vmec_input, [222](#)
- nfort18
 - vparams, [279](#)
- nfp
 - nestor_io, [46](#)
 - read_wout_mod, [124](#)
 - vmec_input, [223](#)
- nfper
 - vacmod0, [205](#)
- nfper0
 - mgrid_mod, [27](#)
- nigroup
 - vsvd0, [281](#)
- niter
 - read_wout_mod, [124](#)
- niter_array
 - vmec_input, [223](#)
- niter_default
 - vmec_input, [223](#)
- niterv
 - vmec_main, [251](#)
- nlim
 - mgrid_mod, [27](#)
- nlim_max
 - mgrid_mod, [28](#)
- nlimset
 - mgrid_mod, [28](#)
- nmercier0
 - vparams, [279](#)
- nnyq
 - read_wout_mod, [125](#)
 - vmec_params, [265](#)
- nobd
 - mgrid_mod, [28](#)
- nobser
 - mgrid_mod, [28](#)
- norm_term_flag
 - vmec_params, [265](#)
- np0b
 - mgrid_mod, [28](#)
- nr0b
 - mgrid_mod, [28](#)
- nrgrid
 - mgrid_mod, [29](#)
- nrzt
 - vmec_dim, [213](#)
- ns
 - read_wout_mod, [125](#)
 - vmec_dim, [213](#)
- ns1
 - vmec_dim, [214](#)
- ns4
 - vmec_params, [265](#)
- ns_array
 - vmec_input, [223](#)
- ns_default
 - vmec_input, [223](#)
- ns_error_flag
 - vmec_params, [265](#)
- ns_maxval
 - vmec_dim, [214](#)
- nscale
 - vmec_params, [266](#)
- nsd
 - vparams, [279](#)

- nsets
 - mgrid_mod, [29](#)
- nsetsn
 - mgrid_mod, [29](#)
- nstep
 - vmec_input, [224](#)
- nstore_seq
 - read_wout_mod, [125](#)
 - vparams, [280](#)
- ntheta
 - nestor_io, [46](#)
 - vmec_input, [224](#)
- ntheta1
 - vmec_dim, [214](#)
- ntheta2
 - vmec_dim, [214](#)
- ntheta3
 - vmec_dim, [214](#)
- nthreed
 - vparams, [280](#)
- nthreed0
 - vparams, [280](#)
- ntmax
 - read_wout_mod, [125](#)
 - vmec_params, [266](#)
- ntor
 - nestor_io, [46](#)
 - read_wout_mod, [125](#)
 - vmec_input, [224](#)
- ntor1
 - vmec_dim, [215](#)
- ntor1d
 - vparams, [280](#)
- ntor_default
 - vmec_input, [224](#)
- ntord
 - vparams, [280](#)
- nu
 - vacmod0, [205](#)
- nu2
 - vacmod0, [205](#)
- nu3
 - vacmod0, [205](#)
- nuv
 - vacmod0, [205](#)
- nuv2
 - vacmod0, [206](#)
- nuv_tan
 - vacmod0, [206](#)
- nv
 - vacmod0, [206](#)
- nvacskip
 - vmec_input, [224](#)
- nvp
 - vacmod0, [206](#)
- nvper
 - vacmod0, [206](#)
- nwout0
 - vparams, [281](#)
- nz0b
 - mgrid_mod, [29](#)
- nzeta
 - nestor_io, [47](#)
 - vmec_input, [225](#)
- nzetadim
 - nestor_io, [47](#)
- nzgrid
 - mgrid_mod, [29](#)
- nznt
 - vmec_dim, [215](#)
- nzntdim
 - nestor_io, [47](#)
- ohs
 - vmec_main, [251](#)
- one
 - stel_constants, [168](#)
- onp
 - vacmod, [191](#)
- onp2
 - vacmod, [192](#)
- open_output_files
 - open_output_files.f90, [380](#)
- open_output_files.f90
 - open_output_files, [380](#)
- osqrt2
 - vparams, [281](#)
- otau
 - vmec_main, [252](#)
- otav
 - vmec_main, [252](#)
- output_flag
 - vmec_params, [266](#)
- overr
 - read_wout_mod, [126](#)
 - vmec_main, [252](#)
- p5
 - vacmod, [192](#)
- parse_extension
 - parse_extension.f, [381](#)
- parse_extension.f
 - parse_extension, [381](#)
- pcurr
 - profile_functions.f, [388](#)
- pcurr_type
 - read_wout_mod, [126](#)
 - vmec_input, [225](#)
- pdamp
 - vmec_params, [266](#)
- pfac
 - read_wout_mod, [126](#)
- pfcspec
 - mgrid_mod, [29](#)
- phi
 - read_wout_mod, [126](#)
 - vmec_main, [252](#)

- phidiam
 - read_wout_mod, 126
- phiedge
 - vmec_input, 225
- phiedge_error_flag
 - vmec_params, 266
- phip
 - read_wout_mod, 127
 - realspace, 163
- phipf
 - read_wout_mod, 127
 - vmec_main, 252
- phips
 - vmec_main, 253
- pi
 - stel_constants, 168
- pi2
 - vacmod, 192
- pi3
 - vacmod, 192
- pi4
 - vacmod, 192
- pio2
 - stel_constants, 169
- piota
 - profile_functions.f, 388
- piota_type
 - read_wout_mod, 127
 - vmec_input, 225
- pknots
 - read_wout_mod, 127
- plbfld
 - mgrid_mod, 30
- plflux
 - mgrid_mod, 30
- pmass
 - profile_functions.f, 389
- pmass_type
 - read_wout_mod, 127
 - vmec_input, 225
- polflux
 - magnetic_fluxes.f90, 350
- polflux_deriv
 - magnetic_fluxes.f90, 351
- potu
 - vacmod, 193
- potv
 - vacmod, 193
- potvac
 - read_wout_mod, 128
 - vacmod, 193
- precal
 - precal.f90, 375
- precal.f90
 - precal, 375
- precal_done
 - vacmod, 193
- precondn
 - precondn.f90, 382
- precondn.f90
 - precondn, 382
- pres
 - read_wout_mod, 128
 - vmec_main, 253
- pres_scale
 - vmec_input, 226
- presf
 - read_wout_mod, 128
 - vmec_main, 253
- presgrad
 - vmec_main, 253
- presmid
 - read_wout_mod, 128
- printout
 - printout.f90, 384
- printout.f90
 - printout, 384
- profil1d
 - profil1d.f90, 385
- profil1d.f90
 - profil1d, 385
- profil3d
 - profil3d.f90, 387
- profil3d.f90
 - profil3d, 387
- profile_functions.f
 - pcurr, 388
 - piota, 388
 - pmass, 389
- psi
 - vmec_main, 253
- psiext
 - mgrid_mod, 30
- qfact
 - read_wout_mod, 128
- qmeas
 - read_wout_mod, 129
- qmids
 - read_wout_mod, 129
- r00
 - vmec_main, 254
- r0m
 - vacmod, 193
- r0p
 - vacmod, 194
- r0scale
 - vmec_main, 254
- r1
 - realspace, 164
- r1b
 - vacmod, 194
- r1m
 - vacmod, 194
- r1p
 - vacmod, 194

r2dim
 nestor_io, 47
 ra1m
 vacmod, 194
 ra1p
 vacmod, 195
 raw_coil_current
 mgrid_mod, 30
 raxis
 nestor_io, 47
 read_wout_mod, 129
 raxis_cc
 vmec_input, 226
 raxis_cs
 vmec_input, 226
 raxis_nestor
 vacmod, 195
 rbc
 vmec_input, 226
 rbcoil
 mgrid_mod, 30
 rbcoilsqr
 mgrid_mod, 31
 rbs
 vmec_input, 226
 rbsq
 vmec_main, 254
 rbtor
 nestor_io, 48
 read_wout_mod, 129
 vmec_main, 254
 rbtor0
 read_wout_mod, 129
 vmec_main, 254
 rcc
 vmec_params, 267
 rcon
 realspace, 164
 rcon0
 realspace, 164
 rcosuv
 vacmod, 195
 rcs
 vmec_params, 267
 read_indata
 read_indata.f90, 390
 read_indata.f90
 read_indata, 390
 read_indata_namelist
 vmec_input, 216
 read_mgrid
 mgrid_mod, 20
 read_nestor_inputs
 nestor_io, 40
 read_nestor_outputs
 nestor_io.f90, 366
 read_wout_deallocate
 read_wout_mod, 80
 read_wout_mod, 71
 ac, 81
 ac_aux_f, 81
 ac_aux_s, 81
 ai, 82
 ai_aux_f, 82
 ai_aux_s, 82
 alfa, 82
 am, 82
 am_aux_f, 83
 am_aux_s, 83
 aminor, 83
 anglemse, 83
 aspect, 83
 b0, 84
 bbc, 84
 bcwgt, 84
 bdotgradv, 84
 beta_vol, 84
 betapol, 85
 betator, 85
 betatot, 85
 betaxis, 85
 bmnc, 85
 bmns, 86
 bsubsmnc, 86
 bsubsmns, 86
 bsubumnc, 86
 bsubumns, 86
 bsubvmnc, 87
 bsubvmns, 87
 bsupumnc, 87
 bsupumns, 87
 bsupvmnc, 87
 bsupvmns, 88
 buco, 88
 bvco, 88
 chi, 88
 chipf, 88
 compute_currents, 79
 curmid, 89
 currumnc, 89
 currumns, 89
 currvmnc, 89
 currvmns, 89
 datastark, 90
 datathom, 90
 dcurr, 90
 delphid, 90
 dgeod, 90
 dmerc, 90
 dshear, 91
 dsiobt, 91
 dwell, 91
 equif, 91
 extcur, 91
 flmwgt, 92
 fsql, 92

fsqr, 92
 fsqt, 92
 fsqz, 92
 ftolv, 93
 gamma, 93
 gmnc, 93
 gmns, 93
 iasym, 93
 ierr_vmec, 94
 imatch_phiedge, 94
 imse, 94
 input_extension, 94
 ionlarmor, 94
 iotaf, 95
 iotas, 95
 ipnodes, 95
 isingn, 95
 isnodes, 95
 itfsq, 96
 itor, 96
 itse, 96
 jcuru, 96
 jcurv, 96
 jdotb, 97
 lasym, 97
 lmnc, 97
 lmns, 97
 ln_abeta, 97
 ln_ac, 98
 ln_ac_aux_f, 98
 ln_ac_aux_s, 98
 ln_actit, 98
 ln_ai, 98
 ln_ai_aux_f, 99
 ln_ai_aux_s, 99
 ln_am, 99
 ln_am_aux_f, 99
 ln_am_aux_s, 99
 ln_amin, 100
 ln_aspect, 100
 ln_asym, 100
 ln_b0, 100
 ln_beta, 100
 ln_betah, 101
 ln_bgrv, 101
 ln_bmnc, 101
 ln_bmns, 101
 ln_bsubsmnc, 101
 ln_bsubsmns, 102
 ln_bsubumnc, 102
 ln_bsubumnc_sur, 102
 ln_bsubumns, 102
 ln_bsubumns_sur, 102
 ln_bsubvmnc, 103
 ln_bsubvmnc_sur, 103
 ln_bsubvmns, 103
 ln_bsubvmns_sur, 103
 ln_bsupumnc, 103
 ln_bsupumnc_sur, 104
 ln_bsupumns, 104
 ln_bsupumns_sur, 104
 ln_bsupvmnc, 104
 ln_bsupvmnc_sur, 104
 ln_bsupvmns, 105
 ln_bsupvmns_sur, 105
 ln_buco, 105
 ln_bvco, 105
 ln_chi, 105
 ln_chipf, 106
 ln_ctor, 106
 ln_curlab, 106
 ln_equip, 106
 ln_error, 106
 ln_extcur, 107
 ln_extension, 107
 ln_fp, 107
 ln_free, 107
 ln_fsq, 107
 ln_fsql, 108
 ln_fsqr, 108
 ln_fsqz, 108
 ln_ftolv, 108
 ln_gam, 108
 ln_gmnc, 108
 ln_gmns, 109
 ln_iotaf, 109
 ln_iotah, 109
 ln_jcuru, 109
 ln_jcurv, 109
 ln_jdotb, 110
 ln_lar, 110
 ln_lmnc, 110
 ln_lmns, 110
 ln_magen, 110
 ln_mass, 111
 ln_maxit, 111
 ln_maxmod, 111
 ln_maxmod_nyq, 111
 ln_maxr, 111
 ln_maxz, 112
 ln_mcurr, 112
 ln_merc, 112
 ln_mgeo, 112
 ln_mgrid, 112
 ln_minr, 113
 ln_modb, 113
 ln_mse, 113
 ln_mshear, 113
 ln_mwell, 113
 ln_pbeta, 114
 ln_pcurr_type, 114
 ln_phi, 114
 ln_phiip, 114
 ln_phiipf, 114
 ln_piota_type, 115
 ln_pmass_type, 115

ln_pmod, 115
ln_pmod_nyq, 115
ln_polmod, 115
ln_potvac, 116
ln_presf, 116
ln_presh, 116
ln_qfact, 116
ln_racc, 116
ln_racs, 117
ln_radnod, 117
ln_rbc, 117
ln_rbs, 117
ln_rbt0, 117
ln_rbt1, 118
ln_recon, 118
ln_rmaj, 118
ln_rmnc, 118
ln_rmns, 118
ln_sgs, 119
ln_specw, 119
ln_tbeta, 119
ln_therm, 119
ln_thom, 119
ln_tmod, 120
ln_tmod_nyq, 120
ln_tormod, 120
ln_version, 120
ln_vol, 120
ln_vp, 121
ln_wdot, 121
ln_zacc, 121
ln_zacs, 121
ln_zbc, 121
ln_zbs, 122
ln_zmnc, 122
ln_zmns, 122
loadrzi, 80
lthreed, 122
lwout_opened, 122
machsq, 123
mass, 123
mgrid_file, 123
mnmax, 123
mnmax_nyq, 123
mnyq, 124
mpol, 124
msewgt, 124
nfp, 124
niter, 124
nnyq, 125
ns, 125
nstore_seq, 125
ntmax, 125
ntor, 125
overr, 126
pcurr_type, 126
pfac, 126
phi, 126
phidiam, 126
phip, 127
phipf, 127
piota_type, 127
pknots, 127
pmass_type, 127
potvac, 128
pres, 128
presf, 128
presmid, 128
qfact, 128
qmeas, 129
qmid, 129
raxis, 129
rbtor, 129
rbtor0, 129
read_wout_deallocate, 80
readw_and_open, 80
rmajor, 130
rmax_surf, 130
rmid, 130
rmin_surf, 130
rmnc, 130
rmns, 131
rstark, 131
rthom, 131
rzi_local, 131
shear, 131
sknots, 132
specw, 132
tosuvspace, 81
tswgt, 132
version_, 132
vn_abeta, 132
vn_ac, 132
vn_ac_aux_f, 133
vn_ac_aux_s, 133
vn_actit, 133
vn_ah, 133
vn_ai, 133
vn_ai_aux_f, 134
vn_ai_aux_s, 134
vn_am, 134
vn_am_aux_f, 134
vn_am_aux_s, 134
vn_amin, 135
vn_aspect, 135
vn_asym, 135
vn_atuname, 135
vn_b0, 135
vn_beta, 136
vn_betah, 136
vn_bgrv, 136
vn_bmnc, 136
vn_bmns, 136
vn_bsubsmnc, 137
vn_bsubsmns, 137
vn_bsubumnc, 137

vn_bsubumnc_sur, 137
 vn_bsubumns, 137
 vn_bsubumns_sur, 138
 vn_bsubvmnc, 138
 vn_bsubvmnc_sur, 138
 vn_bsubvmns, 138
 vn_bsubvmns_sur, 138
 vn_bsupumnc, 139
 vn_bsupumnc_sur, 139
 vn_bsupumns, 139
 vn_bsupumns_sur, 139
 vn_bsupvmnc, 139
 vn_bsupvmnc_sur, 139
 vn_bsupvmns, 140
 vn_bsupvmns_sur, 140
 vn_buco, 140
 vn_bvco, 140
 vn_chi, 140
 vn_chipf, 141
 vn_ctor, 141
 vn_curlab, 141
 vn_equif, 141
 vn_error, 141
 vn_extcur, 142
 vn_extension, 142
 vn_fp, 142
 vn_free, 142
 vn_fsq, 142
 vn_fsqli, 143
 vn_fsqr, 143
 vn_fsqz, 143
 vn_ftolv, 143
 vn_gam, 143
 vn_gmnc, 144
 vn_gmns, 144
 vn_iotaf, 144
 vn_iotah, 144
 vn_jcuru, 144
 vn_jcurv, 145
 vn_jdotb, 145
 vn_lar, 145
 vn_lmnc, 145
 vn_lmns, 145
 vn_magen, 146
 vn_mass, 146
 vn_maxit, 146
 vn_maxmod, 146
 vn_maxmod_nyq, 146
 vn_maxr, 147
 vn_maxz, 147
 vn_mcurr, 147
 vn_merc, 147
 vn_mgeo, 147
 vn_mgrid, 148
 vn_minr, 148
 vn_modb, 148
 vn_mse, 148
 vn_mshear, 148
 vn_mwell, 149
 vn_overr, 149
 vn_pbeta, 149
 vn_pcurr_type, 149
 vn_phi, 149
 vn_phip, 150
 vn_phipf, 150
 vn_piota_type, 150
 vn_pmass_type, 150
 vn_pmod, 150
 vn_pmod_nyq, 151
 vn_polmod, 151
 vn_potvac, 151
 vn_presf, 151
 vn_presh, 151
 vn_qfact, 152
 vn_racc, 152
 vn_racs, 152
 vn_radnod, 152
 vn_rbc, 152
 vn_rbs, 153
 vn_rbt0, 153
 vn_rbt1, 153
 vn_rmaj, 153
 vn_rmnc, 153
 vn_rmns, 154
 vn_sgs, 154
 vn_specw, 154
 vn_tbeta, 154
 vn_therm, 154
 vn_thom, 155
 vn_tmod, 155
 vn_tmod_nyq, 155
 vn_tormod, 155
 vn_version, 155
 vn_vol, 156
 vn_vp, 156
 vn_wdot, 156
 vn_zacc, 156
 vn_zacs, 156
 vn_zbc, 157
 vn_zbs, 157
 vn_zmnc, 157
 vn_zmns, 157
 volavgb, 157
 volume, 158
 vp, 158
 wb, 158
 wdot, 158
 wp, 158
 xm, 159
 xm_nyq, 159
 xn, 159
 xn_nyq, 159
 y2stark, 159
 y2thom, 160
 ystark, 160
 ythom, 160

- zaxis, [160](#)
- zmax_surf, [160](#)
- zmnc, [160](#)
- zmns, [161](#)
- read_wout_mod::read_wout_file, [285](#)
- readw_and_open, [285](#)
- readin
 - readin.f90, [400](#)
- readin.f90
 - readin, [400](#)
- readin_flag
 - vmec_params, [267](#)
- README.md, [289](#)
- readw_and_open
 - read_wout_mod, [80](#)
 - read_wout_mod::read_wout_file, [285](#)
- realspace, [161](#)
 - chip, [162](#)
 - extra1, [162](#)
 - extra2, [162](#)
 - extra3, [162](#)
 - extra4, [162](#)
 - gcon, [163](#)
 - guu, [163](#)
 - guv, [163](#)
 - gvv, [163](#)
 - phip, [163](#)
 - r1, [164](#)
 - rcon, [164](#)
 - rcon0, [164](#)
 - ru, [164](#)
 - ru0, [164](#)
 - rv, [165](#)
 - shalf, [165](#)
 - sqrts, [165](#)
 - wint, [165](#)
 - z1, [166](#)
 - zcon, [166](#)
 - zcon0, [166](#)
 - zu, [166](#)
 - zu0, [166](#)
 - zv, [167](#)
- res0
 - vmec_main, [255](#)
- reset_jacdt_flag
 - vmec_params, [267](#)
- reset_params
 - reset_params.f90, [401](#)
- reset_params.f90
 - reset_params, [401](#)
- residue
 - residue.f90, [403](#)
- residue.f90
 - constrain_m1, [402](#)
 - residue, [403](#)
 - scale_m1, [404](#)
- reslim
 - mgrid_mod, [31](#)
- restart_flag
 - vmec_params, [267](#)
- restart_iter
 - restart_iter.f90, [405](#)
- restart_iter.f90
 - restart_iter, [405](#)
- rinner
 - vmec_main, [255](#)
- rlim
 - mgrid_mod, [31](#)
- rmajor
 - read_wout_mod, [130](#)
- rmajor_p
 - vmec_io, [230](#)
- rmax_surf
 - read_wout_mod, [130](#)
 - vmec_io, [230](#)
- rmaxb
 - mgrid_mod, [31](#)
- rmid
 - read_wout_mod, [130](#)
- rmin_surf
 - read_wout_mod, [130](#)
 - vmec_io, [231](#)
- rminb
 - mgrid_mod, [31](#)
- rmn_bdy
 - vmec_main, [255](#)
- rmnc
 - nestor_io, [48](#)
 - read_wout_mod, [130](#)
- rmns
 - nestor_io, [48](#)
 - read_wout_mod, [131](#)
- router
 - vmec_main, [255](#)
- rprec
 - stel_kinds, [170](#)
- rru_fac
 - fbal, [11](#)
- rsc
 - vmec_params, [268](#)
- rsinuv
 - vacmod, [195](#)
- rss
 - vmec_params, [268](#)
- rstark
 - read_wout_mod, [131](#)
- rthom
 - read_wout_mod, [131](#)
- ru
 - realspace, [164](#)
- ru0
 - realspace, [164](#)
- rub
 - vacmod, [195](#)
- ruu
 - vacmod, [196](#)

- ruv
 - vacmod, 196
- rv
 - realspace, 165
- rvb
 - vacmod, 196
- rvv
 - vacmod, 196
- rx1
 - mgrid_mod, 32
- rx2
 - mgrid_mod, 32
- rz2
 - vacmod, 196
- rzl_local
 - read_wout_mod, 131
- rzu_fac
 - fbal, 12
- safe_open
 - safe_open_mod, 167
- safe_open_mod, 167
 - safe_open, 167
- scale_m1
 - residue.f90, 404
- scalfor
 - scalfor.f90, 406
- scalfor.f90
 - scalfor, 406
- scalpot
 - scalpot.f90, 376
- scalpot.f90
 - scalpot, 376
- scalxc
 - xstuff, 282
- seplim
 - mgrid_mod, 32
- set_nestor_sizes
 - vacmod0, 203
- shalf
 - realspace, 165
- shear
 - read_wout_mod, 131
- sifa
 - vmec_main, 255
- sifa2
 - vmec_main, 256
- signgs
 - nestor_io, 48
 - vmec_params, 268
- sin01
 - vmec_persistent, 272
- sinmu
 - vmec_persistent, 272
- sinmui
 - vmec_persistent, 273
- sinmum
 - vmec_persistent, 273
- sinmumi
 - vmec_persistent, 273
- sinnv
 - vmec_persistent, 273
- sinnvn
 - vmec_persistent, 273
- sinper
 - vac_persistent, 174
- sinu
 - vac_persistent, 174
- sinu1
 - vac_persistent, 174
- sinui
 - vac_persistent, 174
- sinuv
 - vac_persistent, 175
- sinv
 - vac_persistent, 175
- sinv1
 - vac_persistent, 175
- sknots
 - read_wout_mod, 132
- slm
 - vacmod, 197
- slp
 - vacmod, 197
- slpm
 - vacmod, 197
- sm
 - vmec_main, 256
- snr
 - vacmod, 197
- snv
 - vacmod, 197
- snz
 - vacmod, 198
- solver
 - solver.f90, 407
- solver.f90
 - solver, 407
- source
 - vacmod, 198
- sp
 - vmec_main, 256
- spectrum
 - spectrum.f90, 407
- spectrum.f90
 - spectrum, 407
- specw
 - read_wout_mod, 132
 - vmec_main, 256
- spfa
 - vmec_main, 256
- spfa2
 - vmec_main, 257
- spline_akima
 - spline_akima.f, 408
- spline_akima.f
 - spline_akima, 408

spline_akima_int
 spline_akima_int.f, 409
 spline_akima_int.f
 spline_akima_int, 409
 spline_cubic
 spline_cubic.f, 410
 spline_cubic.f
 spline_cubic, 410
 spline_nr, 411
 splint_nr, 411
 spline_cubic_int
 spline_cubic_int.f, 412
 spline_cubic_int.f
 spline_cubic_int, 412
 spline_int, 413
 splint_int, 414
 spline_int
 spline_cubic_int.f, 413
 spline_nr
 spline_cubic.f, 411
 splint_int
 spline_cubic_int.f, 414
 splint_nr
 spline_cubic.f, 411
 spres_ped
 vmec_input, 227
 squad1u
 vacmod, 198
 squad2u
 vacmod, 198
 sqrt2
 stel_constants, 169
 sqrta
 vacmod, 198
 sqrtc
 vacmod, 199
 sqrts
 realspace, 165
 src/add_fluxes.f90, 289
 src/alias.f90, 290
 src/allocate_funct3d.f90, 291
 src/allocate_ns.f90, 292
 src/allocate_nunv.f90, 294
 src/aspectratio.f90, 295
 src/bcovar.f90, 296
 src/bextrema.f90, 297
 src/bss.f90, 298
 src/calc_fbal.f90, 300
 src/CMakeLists.txt, 301
 src/convert.f90, 303
 src/data/CMakeLists.txt, 301
 src/data/fbal.f90, 304
 src/data/realspace.f90, 304
 src/data/stel_constants.f90, 305
 src/data/stel_kinds.f90, 306
 src/data/vforces.f90, 306
 src/data/vmec_dim.f90, 307
 src/data/vmec_input.f90, 307
 src/data/vmec_io.f90, 308
 src/data/vmec_main.f90, 309
 src/data/vmec_params.f90, 312
 src/data/vmec_persistent.f90, 313
 src/data/vmercier.f90, 314
 src/data/vparams.f90, 314
 src/data/vsvd0.f90, 315
 src/data/xstuff.f90, 315
 src/elongation.f90, 316
 src/eqfor.f90, 317
 src/eqsolve.f90, 319
 src/evolve.f90, 320
 src/fileout.f90, 322
 src/fixaray.f90, 325
 src/flip_theta.f90, 326
 src/forces.f90, 327
 src/free_mem_funct3d.f90, 328
 src/free_mem_ns.f90, 329
 src/free_mem_nunv.f90, 330
 src/freeb_data.f90, 331
 src/fsym_fft.f90, 332
 src/fsym_invfft.f90, 334
 src/funct3d.f90, 335
 src/functions.f, 337
 src/getbsubs.f90, 337
 src/getcurmid.f90, 339
 src/getfsq.f90, 340
 src/guess_axis.f90, 341
 src/heading.f90, 342
 src/initialize_radial.f90, 343
 src/interp.f90, 344
 src/jacobian.f90, 346
 src/jxbforce.f90, 346
 src/lamcal.f90, 348
 src/line_segment.f, 349
 src/magnetic_fluxes.f90, 350
 src/mercier.f90, 354
 src/mgrid_mod.f, 355
 src/NESTOR/analysum.f90, 358
 src/NESTOR/analysum2.f90, 358
 src/NESTOR/analyt.f90, 359
 src/NESTOR/becoil.f90, 360
 src/NESTOR/belicu.f90, 361
 src/NESTOR/bextern.f90, 362
 src/NESTOR/CMakeLists.txt, 302
 src/NESTOR/data/CMakeLists.txt, 302
 src/NESTOR/data/ncstor_io.f90, 363
 src/NESTOR/data/vac_persistent.f90, 368
 src/NESTOR/data/vacmod.f90, 369
 src/NESTOR/data/vacmod0.f90, 371
 src/NESTOR/fouri.f90, 372
 src/NESTOR/fourp.f90, 373
 src/NESTOR/greenf.f90, 373
 src/NESTOR/ncstor_main.f90, 374
 src/NESTOR/precal.f90, 375
 src/NESTOR/scalpot.f90, 376
 src/NESTOR/surface.f90, 377
 src/NESTOR/vacuum.f90, 378

- src/open_output_files.f90, 380
- src/parse_extension.f, 381
- src/precondn.f90, 382
- src/printout.f90, 384
- src/profil1d.f90, 385
- src/profil3d.f90, 386
- src/profile_functions.f, 387
- src/read_indata.f90, 390
- src/read_wout_mod.f, 391
- src/readin.f90, 399
- src/reset_params.f90, 401
- src/residue.f90, 402
- src/restart_iter.f90, 405
- src/safe_open_mod.f, 405
- src/scalfor.f90, 406
- src/solver.f90, 407
- src/spectrum.f90, 407
- src/spline_akima.f, 408
- src/spline_akima_int.f, 409
- src/spline_cubic.f, 410
- src/spline_cubic_int.f, 412
- src/symforce.f90, 414
- src/symrzi.f90, 416
- src/tolower.f90, 417
- src/tomnsp.f90, 418
- src/totzsp.f90, 419
- src/tridslv.f90, 423
- src/vmec.f90, 424
- src/wrout.f90, 426
- stel_constants, 168
 - degree, 168
 - mu0, 168
 - one, 168
 - pi, 168
 - pio2, 169
 - sqrt2, 169
 - twopi, 169
 - zero, 169
- stel_kinds, 169
 - cprec, 170
 - dp, 170
 - iprec, 170
 - rprec, 170
- successful_term_flag
 - vmec_params, 268
- sum_bfield
 - mgrid_mod, 21
- surf_area_p
 - vmec_io, 231
- surface
 - surface.f90, 377
- surface.f90
 - surface, 377
- symforce
 - symforce.f90, 414
- symforce.f90
 - symforce, 414
 - symoutput, 415
- symoutput
 - symforce.f90, 415
- symrzi
 - symrzi.f90, 416
- symrzi.f90
 - symrzi, 416
- tanu
 - vac_persistent, 175
- tanu_1d
 - vac_persistent, 175
- tanv
 - vac_persistent, 176
- tanv_1d
 - vac_persistent, 176
- tcon
 - vmec_main, 257
- tcon0
 - vmec_input, 227
- timestep_flag
 - vmec_params, 268
- tlm
 - vacmod, 199
- tlm1
 - vacmod, 199
- tlm2
 - vacmod, 199
- tlp
 - vacmod, 199
- tlp1
 - vacmod, 200
- tlp2
 - vacmod, 200
- tlpm
 - vacmod, 200
- tokid
 - mgrid_mod, 32
- tolower
 - tolower.f90, 417
- tolower.f90
 - tolower, 417
- tomnsp.f90
 - tomnspa, 418
 - tomnsps, 419
- tomnspa
 - tomnsp.f90, 418
- tomnsps
 - tomnsp.f90, 419
- torflux
 - magnetic_fluxes.f90, 352
- torflux_deriv
 - magnetic_fluxes.f90, 353
- tosuvspace
 - read_wout_mod, 81
- totzsp.f90
 - convert_asym, 420
 - convert_sym, 420
 - totzspa, 420
 - totzsps, 421

totzspa
 totzsp.f90, 420
 totzsps
 totzsp.f90, 421
 tridslv
 tridslv.f90, 423
 tridslv.f90
 tridslv, 423
 tswgt
 read_wout_mod, 132
 two
 vacmod, 200
 two_power
 functions, 13
 two_power_gs
 functions, 13
 twopi
 stel_constants, 169
 uminus
 vmec_params, 269
 unpsiext
 mgrid_mod, 32
 vac_persistent, 171
 bsubu_sur, 171
 bsubv_sur, 171
 bsupu_sur, 171
 bsupv_sur, 172
 cmns, 172
 cosper, 172
 cosu, 172
 cosu1, 172
 cosui, 173
 cosuv, 173
 cosv, 173
 cosv1, 173
 csign, 173
 imirr, 174
 sinper, 174
 sinu, 174
 sinu1, 174
 sinui, 174
 sinuv, 175
 sinv, 175
 sinv1, 175
 tanu, 175
 tanu_1d, 175
 tanv, 176
 tanv_1d, 176
 xmpot, 176
 xnpot, 176
 vacmod, 177
 actemp, 180
 adm, 180
 adp, 180
 all_slm, 181
 all_slp, 181
 all_tlm, 181
 all_tlp, 181
 allocate_nestor, 179
 alp, 181
 alu, 182
 alv, 182
 alvp, 182
 amatrix, 182
 amatsav, 182
 astemp, 183
 auu, 183
 auv, 183
 avv, 183
 azm1u, 183
 azp1u, 184
 bcos, 184
 bexn, 184
 bexni, 184
 bexu, 184
 bexv, 185
 bphi, 185
 bphiv, 185
 brad, 185
 brv, 185
 bsin, 186
 bsqvac, 186
 bsubu, 186
 bsubv, 186
 bsubvvac, 186
 bvecsav, 187
 bz, 187
 bzv, 187
 cma, 187
 cma11u, 187
 delt1u, 188
 drv, 188
 dsave, 188
 free_mem_nestor, 179
 g1, 188
 g2, 188
 ga1, 189
 ga2, 189
 green, 189
 greenp, 189
 grpmn, 189
 grpmn_m_map_wrt, 190
 grpmn_n_map_wrt, 190
 gsave, 190
 gstore, 190
 guu_b, 190
 guv_b, 191
 gvv_b, 191
 m_map_wrt, 191
 n_map_wrt, 191
 onp, 191
 onp2, 192
 p5, 192
 pi2, 192
 pi3, 192

- pi4, [192](#)
- potu, [193](#)
- potv, [193](#)
- potvac, [193](#)
- precal_done, [193](#)
- r0m, [193](#)
- r0p, [194](#)
- r1b, [194](#)
- r1m, [194](#)
- r1p, [194](#)
- ra1m, [194](#)
- ra1p, [195](#)
- raxis_nestor, [195](#)
- rcosuv, [195](#)
- rsinuv, [195](#)
- rub, [195](#)
- ruu, [196](#)
- ruv, [196](#)
- rvb, [196](#)
- rvv, [196](#)
- rzb2, [196](#)
- slm, [197](#)
- slp, [197](#)
- slpm, [197](#)
- snr, [197](#)
- snv, [197](#)
- snz, [198](#)
- source, [198](#)
- squad1u, [198](#)
- squad2u, [198](#)
- sqrta, [198](#)
- sqrta, [199](#)
- t1m, [199](#)
- t1m1, [199](#)
- t1m2, [199](#)
- t1p, [199](#)
- t1p1, [200](#)
- t1p2, [200](#)
- t1pm, [200](#)
- two, [200](#)
- xpts, [200](#)
- z1b, [201](#)
- zaxis_nestor, [201](#)
- zub, [201](#)
- zuu, [201](#)
- zuv, [201](#)
- zvb, [202](#)
- zvv, [202](#)
- vacmod0, [202](#)
- mf, [203](#)
- mf1, [203](#)
- mnpd, [204](#)
- mnpd2, [204](#)
- ndim, [204](#)
- nf, [204](#)
- nf1, [204](#)
- nfper, [205](#)
- nu, [205](#)
- nu2, [205](#)
- nu3, [205](#)
- nuv, [205](#)
- nuv2, [206](#)
- nuv_tan, [206](#)
- nv, [206](#)
- nvp, [206](#)
- nvper, [206](#)
- set_nestor_sizes, [203](#)
- vacuum
 - vacuum.f90, [378](#)
- vacuum.f90
 - vacuum, [378](#)
- vacuum_calls
 - nestor_io, [48](#)
 - vmec_main, [257](#)
- version_
 - read_wout_mod, [132](#)
 - vmec_params, [269](#)
- vforces, [207](#)
 - armn, [207](#)
 - armn_e, [207](#)
 - armn_o, [208](#)
 - azmn, [208](#)
 - azmn_e, [208](#)
 - azmn_o, [208](#)
 - blmn, [208](#)
 - blmn_e, [209](#)
 - blmn_o, [209](#)
 - brmn, [209](#)
 - brmn_e, [209](#)
 - brmn_o, [209](#)
 - bzmn, [210](#)
 - bzmn_e, [210](#)
 - bzmn_o, [210](#)
 - clmn, [210](#)
 - clmn_e, [210](#)
 - clmn_o, [211](#)
 - crmn, [211](#)
 - crmn_e, [211](#)
 - crmn_o, [211](#)
 - czmn, [211](#)
 - czmn_e, [212](#)
 - czmn_o, [212](#)
- vmec
 - vmec.f90, [424](#)
- vmec.f90
 - vmec, [424](#)
 - xvmec, [425](#)
- vmec_dim, [212](#)
 - mnmax, [212](#)
 - mnmax_nyq, [213](#)
 - mns, [213](#)
 - mnsiz, [213](#)
 - mpol1, [213](#)
 - nrzt, [213](#)
 - ns, [213](#)
 - ns1, [214](#)

ns_maxval, 214
ntheta1, 214
ntheta2, 214
ntheta3, 214
ntor1, 215
nznt, 215
vmec_input, 215
 ac, 217
 ac_aux_f, 217
 ac_aux_s, 218
 ai, 218
 ai_aux_f, 218
 ai_aux_s, 218
 am, 218
 am_aux_f, 219
 am_aux_s, 219
 aphi, 219
 bloat, 219
 curtor, 219
 delt, 220
 extcur, 220
 ftol_array, 220
 ftol_default, 220
 gamma, 220
 input_extension, 221
 lasym, 221
 lbsubs, 221
 lfreeb, 221
 mfilter_fbdy, 221
 mgrid_file, 222
 mpol, 222
 mpol_default, 222
 ncurr, 222
 nfilter_fbdy, 222
 nfp, 223
 niter_array, 223
 niter_default, 223
 ns_array, 223
 ns_default, 223
 nstep, 224
 ntheta, 224
 ntor, 224
 ntor_default, 224
 nvacskip, 224
 nzeta, 225
 pcurr_type, 225
 phiedge, 225
 piota_type, 225
 pmass_type, 225
 pres_scale, 226
 raxis_cc, 226
 raxis_cs, 226
 rbc, 226
 rbs, 226
 read_indata_namelist, 216
 spres_ped, 227
 tcon0, 227
 write_indata_namelist, 217
 zaxis_cc, 227
 zaxis_cs, 227
 zbc, 227
 zbs, 228
vmec_io, 228
 aminor_p, 228
 b0, 228
 betapol, 229
 betator, 229
 betatot, 229
 betaxis, 229
 circum_p, 229
 cross_area_p, 230
 ionlarmor, 230
 kappa_p, 230
 rmajor_p, 230
 rmax_surf, 230
 rmin_surf, 231
 surf_area_p, 231
 volavgb, 231
 volume_p, 231
 zmax_surf, 231
vmec_main, 232
 ard, 235
 arm, 235
 aspect, 235
 azd, 235
 azm, 235
 bdamp, 235
 bdotb, 236
 bdotgradv, 236
 beta_vol, 236
 blam, 236
 bmax, 236
 bmin, 237
 bpedge, 237
 brd, 237
 bredg, 237
 brm, 237
 bsqsav, 238
 bsubu0, 238
 buc0, 238
 buc0f, 238
 bvco, 238
 bvcof, 239
 bzd, 239
 bzedg, 239
 bzm, 239
 chi, 239
 chipf, 240
 chips, 240
 clam, 240
 crd, 240
 ctor, 240
 currv, 241
 dbsq, 241
 delbsq, 241
 delt0r, 241

dlam, 241
 equif, 242
 faccon, 242
 faclam, 242
 faclam0, 242
 fedge, 242
 fnorm, 243
 fnorm1, 243
 fnorml, 243
 fsq, 243
 fsql, 243
 fsql1, 244
 fsqr, 244
 fsqr1, 244
 fsqsum0, 244
 fsqz, 244
 fsqz1, 245
 ftolv, 245
 hi, 245
 hp, 245
 hs, 245
 icurv, 246
 iequi, 246
 ijacob, 246
 iotaf, 246
 iotas, 247
 ireflect, 247
 irst, 247
 irzloff, 247
 iter1, 248
 iter2, 248
 itfsq, 248
 ivac, 248
 jcuru, 249
 jcurv, 249
 jdotb, 249
 jpar2, 249
 jperp2, 249
 lconm1, 250
 lflip, 250
 lthreed, 250
 mass, 250
 multi_ns_grid, 250
 ndatai, 251
 ndatap, 251
 neqs, 251
 niterv, 251
 ohs, 251
 otau, 252
 otav, 252
 overr, 252
 phi, 252
 phipf, 252
 phips, 253
 pres, 253
 presf, 253
 presgrad, 253
 psi, 253
 r00, 254
 r0scale, 254
 rbsq, 254
 rbtor, 254
 rbtor0, 254
 res0, 255
 rinner, 255
 rmn_bdy, 255
 router, 255
 sifa, 255
 sifa2, 256
 sm, 256
 sp, 256
 specw, 256
 spfa, 256
 spfa2, 257
 tcon, 257
 vacuum_calls, 257
 voli, 257
 vp, 257
 vpphi, 258
 wb, 258
 wp, 258
 xcl0, 258
 xmpq, 258
 yellip, 259
 ygeo, 259
 yinden, 259
 yshift, 259
 ytrian, 259
 z00, 260
 zmn_bdy, 260
 vmec_params, 260
 bad_jacobian_flag, 261
 cleanup_flag, 261
 input_error_flag, 262
 jac75_flag, 262
 jlam, 262
 jmin1, 262
 jmin2, 262
 lamscale, 263
 m0, 263
 m1, 263
 meven, 263
 misc_error_flag, 264
 mnyq, 264
 modd, 264
 mscale, 264
 n0, 264
 ndamp, 265
 nnyq, 265
 norm_term_flag, 265
 ns4, 265
 ns_error_flag, 265
 nscale, 266
 ntmax, 266
 output_flag, 266
 pdamp, 266

phiedge_error_flag, 266
 rcc, 267
 rcs, 267
 readin_flag, 267
 reset_jacdt_flag, 267
 restart_flag, 267
 rsc, 268
 rss, 268
 signgs, 268
 successful_term_flag, 268
 timestep_flag, 268
 uminus, 269
 version_, 269
 zcc, 269
 zcs, 269
 zsc, 269
 zss, 270
 vmec_persistent, 270
 cos01, 270
 cosmu, 271
 cosmui, 271
 cosmui3, 271
 cosmum, 271
 cosmumi, 271
 cosmumi3, 271
 cosnv, 272
 cosnvn, 272
 ixm, 272
 jmin3, 272
 sin01, 272
 sinmu, 272
 sinmui, 273
 sinmum, 273
 sinmumi, 273
 sinnv, 273
 sinnav, 273
 xm, 274
 xm_nyq, 274
 xn, 274
 xn_nyq, 274
 vmecier, 275
 dcurr, 275
 dgeod, 275
 dmerc, 275
 dshear, 275
 dwell, 276
 vn_abeta
 read_wout_mod, 132
 vn_ac
 read_wout_mod, 132
 vn_ac_aux_f
 read_wout_mod, 133
 vn_ac_aux_s
 read_wout_mod, 133
 vn_actit
 read_wout_mod, 133
 vn_adm
 nestor_io, 49
 vn_adp
 nestor_io, 49
 vn_ah
 read_wout_mod, 133
 vn_ai
 read_wout_mod, 133
 vn_ai_aux_f
 read_wout_mod, 134
 vn_ai_aux_s
 read_wout_mod, 134
 vn_all_slm
 nestor_io, 49
 vn_all_slp
 nestor_io, 49
 vn_all_tlm
 nestor_io, 49
 vn_all_tlp
 nestor_io, 50
 vn_am
 read_wout_mod, 134
 vn_am_aux_f
 read_wout_mod, 134
 vn_am_aux_s
 read_wout_mod, 134
 vn_amatrix
 nestor_io, 50
 vn_amatsav
 nestor_io, 50
 vn_amin
 read_wout_mod, 135
 vn_aspect
 read_wout_mod, 135
 vn_asym
 read_wout_mod, 135
 vn_atuname
 read_wout_mod, 135
 vn_auu
 nestor_io, 50
 vn_auv
 nestor_io, 50
 vn_avv
 nestor_io, 51
 vn_azm1u
 nestor_io, 51
 vn_azp1u
 nestor_io, 51
 vn_b0
 read_wout_mod, 135
 vn_beta
 read_wout_mod, 136
 vn_betah
 read_wout_mod, 136
 vn_bexn
 nestor_io, 51
 vn_bexni
 nestor_io, 51
 vn_bexu
 nestor_io, 52

vn_bexv
 nestor_io, [52](#)
vn_bgrv
 read_wout_mod, [136](#)
vn_bmnc
 read_wout_mod, [136](#)
vn_bmns
 read_wout_mod, [136](#)
vn_bp0
 mgrid_mod, [32](#)
vn_bphi
 nestor_io, [52](#)
vn_bphiv
 nestor_io, [52](#)
vn_br0
 mgrid_mod, [33](#)
vn_brad
 nestor_io, [52](#)
vn_brv
 nestor_io, [53](#)
vn_bsqvac
 nestor_io, [53](#)
vn_bsubsmnc
 read_wout_mod, [137](#)
vn_bsubsmns
 read_wout_mod, [137](#)
vn_bsubu
 nestor_io, [53](#)
vn_bsubumnc
 read_wout_mod, [137](#)
vn_bsubumnc_sur
 read_wout_mod, [137](#)
vn_bsubumns
 read_wout_mod, [137](#)
vn_bsubumns_sur
 read_wout_mod, [138](#)
vn_bsubv
 nestor_io, [53](#)
vn_bsubvmnc
 read_wout_mod, [138](#)
vn_bsubvmnc_sur
 read_wout_mod, [138](#)
vn_bsubvmns
 read_wout_mod, [138](#)
vn_bsubvmns_sur
 read_wout_mod, [138](#)
vn_bsubvvac
 nestor_io, [53](#)
vn_bsupumnc
 read_wout_mod, [139](#)
vn_bsupumnc_sur
 read_wout_mod, [139](#)
vn_bsupumns
 read_wout_mod, [139](#)
vn_bsupumns_sur
 read_wout_mod, [139](#)
vn_bsupvmnc
 read_wout_mod, [139](#)
vn_bsupvmnc_sur
 read_wout_mod, [139](#)
vn_bsupvmns
 read_wout_mod, [140](#)
vn_bsupvmns_sur
 read_wout_mod, [140](#)
vn_buco
 read_wout_mod, [140](#)
vn_bvco
 read_wout_mod, [140](#)
vn_bvecsav
 nestor_io, [54](#)
vn_bz
 nestor_io, [54](#)
vn_bz0
 mgrid_mod, [33](#)
vn_bzv
 nestor_io, [54](#)
vn_chi
 read_wout_mod, [140](#)
vn_chipf
 read_wout_mod, [141](#)
vn_cma
 nestor_io, [54](#)
vn_cma11u
 nestor_io, [54](#)
vn_coilcur
 mgrid_mod, [33](#)
vn_coilgrp
 mgrid_mod, [33](#)
vn_ctor
 nestor_io, [55](#)
 read_wout_mod, [141](#)
vn_curlab
 read_wout_mod, [141](#)
vn_delt1u
 nestor_io, [55](#)
vn_drv
 nestor_io, [55](#)
vn_equip
 read_wout_mod, [141](#)
vn_error
 read_wout_mod, [141](#)
vn_extcur
 nestor_io, [55](#)
 read_wout_mod, [142](#)
vn_extension
 read_wout_mod, [142](#)
vn_fp
 read_wout_mod, [142](#)
vn_free
 read_wout_mod, [142](#)
vn_fsq
 read_wout_mod, [142](#)
vn_fsqli
 read_wout_mod, [143](#)
vn_fsqr
 read_wout_mod, [143](#)

vn_fsqz
 read_wout_mod, 143
vn_ftolv
 read_wout_mod, 143
vn_gam
 read_wout_mod, 143
vn_gmnc
 read_wout_mod, 144
vn_gmns
 read_wout_mod, 144
vn_green
 nestor_io, 55
vn_greenp
 nestor_io, 56
vn_grpmn
 nestor_io, 56
vn_grpmn_m_map
 nestor_io, 56
vn_grpmn_n_map
 nestor_io, 56
vn_gstore
 nestor_io, 56
vn_guu_b
 nestor_io, 57
vn_guv_b
 nestor_io, 57
vn_gvv_b
 nestor_io, 57
vn_ier_flag
 nestor_io, 57
vn_imirr
 nestor_io, 57
vn_inputext
 nestor_io, 58
vn_iotaf
 read_wout_mod, 144
vn_iotah
 read_wout_mod, 144
vn_ir
 mgrid_mod, 33
vn_ivac
 nestor_io, 58
vn_ivacskip
 nestor_io, 58
vn_jcuru
 read_wout_mod, 144
vn_jcurv
 read_wout_mod, 145
vn_jdotb
 read_wout_mod, 145
vn_jz
 mgrid_mod, 34
vn_kp
 mgrid_mod, 34
vn_lar
 read_wout_mod, 145
vn_lasym
 nestor_io, 58
vn_lmnc
 read_wout_mod, 145
vn_lmns
 read_wout_mod, 145
vn_m_map
 nestor_io, 58
vn_magen
 read_wout_mod, 146
vn_mass
 read_wout_mod, 146
vn_maxit
 read_wout_mod, 146
vn_maxmod
 read_wout_mod, 146
vn_maxmod_nyq
 read_wout_mod, 146
vn_maxr
 read_wout_mod, 147
vn_maxz
 read_wout_mod, 147
vn_mcurr
 read_wout_mod, 147
vn_merc
 read_wout_mod, 147
vn_mgeo
 read_wout_mod, 147
vn_mgmode
 mgrid_mod, 34
vn_mgrid
 nestor_io, 59
 read_wout_mod, 148
vn_minr
 read_wout_mod, 148
vn_mnmax
 nestor_io, 59
vn_mnpd
 nestor_io, 59
vn_mnpd2
 nestor_io, 59
vn_modb
 read_wout_mod, 148
vn_mpol
 nestor_io, 59
vn_mse
 read_wout_mod, 148
vn_mshear
 read_wout_mod, 148
vn_mwell
 read_wout_mod, 149
vn_n_map
 nestor_io, 60
vn_nextcur
 mgrid_mod, 34
vn_nfp
 mgrid_mod, 34
 nestor_io, 60
vn_ntheta
 nestor_io, 60

vn_ntor
 nestor_io, 60
 vn_nzeta
 nestor_io, 60
 vn_overr
 read_wout_mod, 149
 vn_pbeta
 read_wout_mod, 149
 vn_pcurr_type
 read_wout_mod, 149
 vn_phi
 read_wout_mod, 149
 vn_phip
 read_wout_mod, 150
 vn_phipf
 read_wout_mod, 150
 vn_piota_type
 read_wout_mod, 150
 vn_pmass_type
 read_wout_mod, 150
 vn_pmod
 nestor_io, 61
 read_wout_mod, 150
 vn_pmod_nyq
 read_wout_mod, 151
 vn_polmod
 read_wout_mod, 151
 vn_potu
 nestor_io, 61
 vn_potv
 nestor_io, 61
 vn_potvac
 nestor_io, 61
 read_wout_mod, 151
 vn_presf
 read_wout_mod, 151
 vn_presh
 read_wout_mod, 151
 vn_qfact
 read_wout_mod, 152
 vn_r0m
 nestor_io, 61
 vn_r0p
 nestor_io, 62
 vn_r1b
 nestor_io, 62
 vn_r1m
 nestor_io, 62
 vn_r1p
 nestor_io, 62
 vn_ra1m
 nestor_io, 62
 vn_ra1p
 nestor_io, 63
 vn_racc
 read_wout_mod, 152
 vn_racs
 read_wout_mod, 152
 vn_radnod
 read_wout_mod, 152
 vn_raxis_nestor
 nestor_io, 63
 vn_rbc
 read_wout_mod, 152
 vn_rbs
 read_wout_mod, 153
 vn_rbt0
 read_wout_mod, 153
 vn_rbt1
 read_wout_mod, 153
 vn_rbtor
 nestor_io, 63
 vn_rcosuv
 nestor_io, 63
 vn_rmaj
 read_wout_mod, 153
 vn_rmax
 mgrid_mod, 35
 vn_rmin
 mgrid_mod, 35
 vn_rmnc
 nestor_io, 63
 read_wout_mod, 153
 vn_rmns
 nestor_io, 64
 read_wout_mod, 154
 vn_rsinuv
 nestor_io, 64
 vn_rub
 nestor_io, 64
 vn_ruu
 nestor_io, 64
 vn_ruv
 nestor_io, 64
 vn_rvb
 nestor_io, 65
 vn_rvv
 nestor_io, 65
 vn_rzb2
 nestor_io, 65
 vn_sgs
 read_wout_mod, 154
 vn_signgs
 nestor_io, 65
 vn_snr
 nestor_io, 65
 vn_snv
 nestor_io, 66
 vn_snz
 nestor_io, 66
 vn_specw
 read_wout_mod, 154
 vn_sqad1u
 nestor_io, 66
 vn_sqad2u
 nestor_io, 66

vn_sqrt
 nestor_io, 66
 vn_sqrtc
 nestor_io, 67
 vn_tanu
 nestor_io, 67
 vn_tanv
 nestor_io, 67
 vn_tbeta
 read_wout_mod, 154
 vn_therm
 read_wout_mod, 154
 vn_thom
 read_wout_mod, 155
 vn_tmod
 nestor_io, 67
 read_wout_mod, 155
 vn_tmod_nyq
 read_wout_mod, 155
 vn_tormod
 read_wout_mod, 155
 vn_vacuum_calls
 nestor_io, 67
 vn_version
 read_wout_mod, 155
 vn_vol
 read_wout_mod, 156
 vn_vp
 read_wout_mod, 156
 vn_wdot
 read_wout_mod, 156
 vn_wint
 nestor_io, 68
 vn_xmpot
 nestor_io, 68
 vn_xnpot
 nestor_io, 68
 vn_z1b
 nestor_io, 68
 vn_zacc
 read_wout_mod, 156
 vn_zacs
 read_wout_mod, 156
 vn_zaxis_nestor
 nestor_io, 68
 vn_zbc
 read_wout_mod, 157
 vn_zbs
 read_wout_mod, 157
 vn_zmax
 mgrid_mod, 35
 vn_zmin
 mgrid_mod, 35
 vn_zmnc
 nestor_io, 69
 read_wout_mod, 157
 vn_zmns
 nestor_io, 69
 read_wout_mod, 157
 vn_zub
 nestor_io, 69
 vn_zuu
 nestor_io, 69
 vn_zuv
 nestor_io, 69
 vn_zvb
 nestor_io, 70
 vn_zvv
 nestor_io, 70
 volavgb
 read_wout_mod, 157
 vmec_io, 231
 voli
 vmec_main, 257
 volume
 read_wout_mod, 158
 volume_p
 vmec_io, 231
 vp
 read_wout_mod, 158
 vmec_main, 257
 vparams, 276
 c1pm13, 277
 c1pm2, 277
 c1pm8, 277
 c2p0, 277
 c3p0, 277
 cbig, 277
 cp05, 278
 cp15, 278
 cp25, 278
 cp5, 278
 indata0, 278
 jxbout0, 278
 mpol1d, 279
 mpold, 279
 ndatafmax, 279
 nfort18, 279
 nmercier0, 279
 nsd, 279
 nstore_seq, 280
 nthreed, 280
 nthreed0, 280
 ntor1d, 280
 ntord, 280
 nwout0, 281
 osqrt2, 281
 vpphi
 vmec_main, 258
 vsvd0, 281
 nigroup, 281
 wb
 read_wout_mod, 158
 vmec_main, 258
 wdot
 read_wout_mod, 158

wint
 nestor_io, 70
 realspace, 165
 wp
 read_wout_mod, 158
 vmec_main, 258
 write_indata_namelist
 vmec_input, 217
 write_nestor_inputs
 nestor_io.f90, 367
 write_nestor_outputs
 nestor_io, 41
 wrout
 wrout.f90, 426
 wrout.f90
 wrout, 426
 xc
 xstuff, 282
 xcdot
 xstuff, 282
 xcl0
 vmec_main, 258
 xm
 nestor_io, 70
 read_wout_mod, 159
 vmec_persistent, 274
 xm_nyq
 read_wout_mod, 159
 vmec_persistent, 274
 xmpot
 vac_persistent, 176
 xmpq
 vmec_main, 258
 xn
 nestor_io, 70
 read_wout_mod, 159
 vmec_persistent, 274
 xn_nyq
 read_wout_mod, 159
 vmec_persistent, 274
 xnpot
 vac_persistent, 176
 xobser
 mgrid_mod, 35
 xobsqr
 mgrid_mod, 36
 xpts
 vacmod, 200
 xsave
 xstuff, 283
 xstore
 xstuff, 283
 xstuff, 282
 gc, 282
 scalxc, 282
 xc, 282
 xcdot, 282
 xsave, 283
 xstore, 283
 xvmec
 vmec.f90, 425
 y2stark
 read_wout_mod, 159
 y2thom
 read_wout_mod, 160
 yellip
 vmec_main, 259
 ygeo
 vmec_main, 259
 yinden
 vmec_main, 259
 yshift
 vmec_main, 259
 ystark
 read_wout_mod, 160
 ythom
 read_wout_mod, 160
 ytrian
 vmec_main, 259
 z00
 vmec_main, 260
 z1
 realspace, 166
 z1b
 vacmod, 201
 zaxis
 nestor_io, 71
 read_wout_mod, 160
 zaxis_cc
 vmec_input, 227
 zaxis_cs
 vmec_input, 227
 zaxis_nestor
 vacmod, 201
 zbc
 vmec_input, 227
 zbcoil
 mgrid_mod, 36
 zbs
 vmec_input, 228
 zcc
 vmec_params, 269
 zcon
 realspace, 166
 zcon0
 realspace, 166
 zcs
 vmec_params, 269
 zero
 stel_constants, 169
 zlim
 mgrid_mod, 36
 zmax_surf
 read_wout_mod, 160
 vmec_io, 231

zmaxb
 mgrid_mod, [36](#)
zminb
 mgrid_mod, [36](#)
zmn_bdy
 vmec_main, [260](#)
zmnc
 nestor_io, [71](#)
 read_wout_mod, [160](#)
zmns
 nestor_io, [71](#)
 read_wout_mod, [161](#)
zobser
 mgrid_mod, [37](#)
zsc
 vmec_params, [269](#)
zss
 vmec_params, [270](#)
zu
 realspace, [166](#)
zu0
 realspace, [166](#)
zub
 vacmod, [201](#)
zuu
 vacmod, [201](#)
zuv
 vacmod, [201](#)
zv
 realspace, [167](#)
zvb
 vacmod, [202](#)
zvv
 vacmod, [202](#)
zy1
 mgrid_mod, [37](#)
zy2
 mgrid_mod, [37](#)