

VMEC

8.52

Generated by Doxygen 1.9.1

Wed Apr 28 2021 10:53:41



<b>1 Data Type Index</b>	<b>1</b>
1.1 Data Types List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Data Type Documentation</b>	<b>7</b>
3.1 read_wout_mod::read_wout_file Interface Reference	7
3.1.1 Detailed Description	7
<b>4 File Documentation</b>	<b>9</b>
4.1 src/add_fluxes.f90 File Reference	9
4.2 src/alias.f90 File Reference	9
4.3 src/allocate_func3d.f90 File Reference	9
4.4 src/allocate_ns.f90 File Reference	9
4.5 src/allocate_nunv.f90 File Reference	9
4.6 src/aspectratio.f90 File Reference	10
4.7 src/bcovar.f90 File Reference	10
4.7.1 Function/Subroutine Documentation	10
4.7.1.1 bcovar()	10
4.8 src/bextrema.f90 File Reference	10
4.9 src/bss.f90 File Reference	11
4.10 src/calc_fbal.f90 File Reference	11
4.11 src/convert.f90 File Reference	11
4.12 src/data/fbal.f90 File Reference	11
4.13 src/data/realspace.f90 File Reference	11
4.14 src/data/stel_constants.f File Reference	12
4.15 src/data/stel_kinds.f File Reference	12
4.16 src/data/vforces.f90 File Reference	13
4.17 src/data/vmec_dim.f90 File Reference	13
4.18 src/data/vmec_input.f90 File Reference	14
4.19 src/data/vmec_io.f90 File Reference	15
4.20 src/data/vmec_main.f90 File Reference	15
4.21 src/data/vmec_params.f90 File Reference	18
4.22 src/data/vmec_persistent.f90 File Reference	19
4.23 src/data/vmercier.f90 File Reference	20
4.24 src/data/vparams.f90 File Reference	20
4.25 src/data/vsvd0.f90 File Reference	21
4.26 src/data/xstuff.f90 File Reference	21
4.27 src/elongation.f90 File Reference	21
4.28 src/eqfor.f90 File Reference	21
4.29 src/eqsolve.f90 File Reference	22
4.30 src/evolve.f90 File Reference	22

4.31 src/fileout.f90 File Reference . . . . .	22
4.32 src/fixaray.f90 File Reference . . . . .	22
4.33 src/flip_theta.f90 File Reference . . . . .	22
4.33.1 Function/Subroutine Documentation . . . . .	22
4.33.1.1 flip_theta() . . . . .	22
4.34 src/forces.f90 File Reference . . . . .	23
4.35 src/free_mem_func3d.f90 File Reference . . . . .	23
4.36 src/free_mem_ns.f90 File Reference . . . . .	23
4.37 src/free_mem_nunv.f90 File Reference . . . . .	23
4.38 src/freeb_data.f90 File Reference . . . . .	23
4.39 src/fsym_fft.f90 File Reference . . . . .	23
4.40 src/fsym_invfft.f90 File Reference . . . . .	24
4.41 src/func3d.f90 File Reference . . . . .	24
4.41.1 Function/Subroutine Documentation . . . . .	24
4.41.1.1 func3d() . . . . .	24
4.42 src/functions.f File Reference . . . . .	24
4.43 src/getbsubs.f90 File Reference . . . . .	25
4.44 src/getcurmid.f90 File Reference . . . . .	25
4.45 src/getfsq.f90 File Reference . . . . .	25
4.46 src/guess_axis.f90 File Reference . . . . .	25
4.47 src/heading.f90 File Reference . . . . .	25
4.48 src/initialize_radial.f90 File Reference . . . . .	25
4.49 src/interp.f90 File Reference . . . . .	25
4.50 src/jacobian.f90 File Reference . . . . .	26
4.51 src/jxbforce.f90 File Reference . . . . .	26
4.52 src/lamcal.f90 File Reference . . . . .	26
4.53 src/line_segment.f File Reference . . . . .	26
4.54 src/magnetic_fluxes.f90 File Reference . . . . .	26
4.54.1 Function/Subroutine Documentation . . . . .	26
4.54.1.1 polflux() . . . . .	26
4.54.1.2 polflux_deriv() . . . . .	27
4.54.1.3 torflux() . . . . .	28
4.54.1.4 torflux_deriv() . . . . .	29
4.55 src/mercier.f90 File Reference . . . . .	29
4.56 src/mgrid_mod.f File Reference . . . . .	29
4.57 src/NESTOR/analysum.f90 File Reference . . . . .	31
4.58 src/NESTOR/analysum2.f90 File Reference . . . . .	31
4.59 src/NESTOR/analyt.f90 File Reference . . . . .	31
4.60 src/NESTOR/becoil.f90 File Reference . . . . .	31
4.61 src/NESTOR/belicu.f90 File Reference . . . . .	32
4.62 src/NESTOR/bextern.f90 File Reference . . . . .	32
4.63 src/NESTOR/data/vac_persistent.f90 File Reference . . . . .	32

4.64 src/NESTOR/data/vacmod.f90 File Reference . . . . .	32
4.65 src/NESTOR/data/vacmod0.f90 File Reference . . . . .	35
4.66 src/NESTOR/fouri.f90 File Reference . . . . .	35
4.66.1 Function/Subroutine Documentation . . . . .	35
4.66.1.1 fouri() . . . . .	35
4.67 src/NESTOR/fourp.f90 File Reference . . . . .	36
4.68 src/NESTOR/greenf.f90 File Reference . . . . .	36
4.69 src/NESTOR/precal.f90 File Reference . . . . .	36
4.70 src/NESTOR/scalpot.f90 File Reference . . . . .	36
4.71 src/NESTOR/surface.f90 File Reference . . . . .	36
4.72 src/NESTOR/vacuum.f90 File Reference . . . . .	36
4.73 src/open_output_files.f90 File Reference . . . . .	36
4.74 src/parse_extension.f File Reference . . . . .	37
4.75 src/precondn.f90 File Reference . . . . .	37
4.76 src/printout.f90 File Reference . . . . .	37
4.77 src/profil1d.f90 File Reference . . . . .	37
4.78 src/profil3d.f90 File Reference . . . . .	37
4.79 src/profile_functions.f File Reference . . . . .	37
4.80 src/read_indata.f90 File Reference . . . . .	38
4.81 src/read_wout_mod.f File Reference . . . . .	38
4.82 src/readin.f90 File Reference . . . . .	46
4.83 src/reset_params.f90 File Reference . . . . .	46
4.83.1 Function/Subroutine Documentation . . . . .	46
4.83.1.1 reset_params() . . . . .	46
4.84 src/residue.f90 File Reference . . . . .	46
4.85 src/restart_iter.f90 File Reference . . . . .	46
4.86 src/safe_open_mod.f File Reference . . . . .	47
4.87 src/scalfor.f90 File Reference . . . . .	47
4.88 src/solver.f90 File Reference . . . . .	47
4.89 src/spectrum.f90 File Reference . . . . .	47
4.90 src/spline_akima.f File Reference . . . . .	47
4.91 src/spline_akima_int.f File Reference . . . . .	47
4.92 src/spline_cubic.f File Reference . . . . .	47
4.93 src/spline_cubic_int.f File Reference . . . . .	48
4.94 src/symforce.f90 File Reference . . . . .	48
4.95 src/symrzi.f90 File Reference . . . . .	48
4.96 src/tolower.f90 File Reference . . . . .	48
4.97 src/tomnsp.f90 File Reference . . . . .	48
4.98 src/totzsp.f90 File Reference . . . . .	48
4.98.1 Function/Subroutine Documentation . . . . .	49
4.98.1.1 totzsps() . . . . .	49
4.99 src/tridslv.f90 File Reference . . . . .	49

4.100 src/vmec.f90 File Reference . . . . .	50
4.101 src/wrout.f90 File Reference . . . . .	50
<b>Index</b>	<b>51</b>

# Chapter 1

## Data Type Index

### 1.1 Data Types List

Here are the data types with brief descriptions:

<a href="#">read_wout_mod::read_wout_file</a> . . . . .	7
---	---





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

src/add_fluxes.f90	9
src/alias.f90	9
src/allocate_func3d.f90	9
src/allocate_ns.f90	9
src/allocate_nunv.f90	9
src/aspectratio.f90	10
src/bcovar.f90	10
src/bextrema.f90	10
src/bss.f90	11
src/calc_fbal.f90	11
src/convert.f90	11
src/elongation.f90	21
src/eqfor.f90	21
src/eqsolve.f90	22
src/evolve.f90	22
src/fileout.f90	22
src/fixaray.f90	22
src/flip_theta.f90	22
src/forces.f90	23
src/free_mem_func3d.f90	23
src/free_mem_ns.f90	23
src/free_mem_nunv.f90	23
src/freeb_data.f90	23
src/fsym_fft.f90	23
src/fsym_invfft.f90	24
src/func3d.f90	24
src/functions.f	24
src/getbsubs.f90	25
src/getcurmid.f90	25
src/getfsq.f90	25
src/guess_axis.f90	25
src/heading.f90	25
src/initialize_radial.f90	25
src/interp.f90	25
src/jacobian.f90	26

src/jxbforce.f90	26
src/lamcal.f90	26
src/line_segment.f	26
src/magnetic_fluxes.f90	26
src/mercier.f90	29
src/mgrid_mod.f	29
src/open_output_files.f90	36
src/parse_extension.f	37
src/precondn.f90	37
src/printout.f90	37
src/profil1d.f90	37
src/profil3d.f90	37
src/profile_functions.f	37
src/read_indata.f90	38
src/read_wout_mod.f	38
src/readin.f90	46
src/reset_params.f90	46
src/residue.f90	46
src/restart_iter.f90	46
src/safe_open_mod.f	47
src/scalfor.f90	47
src/solver.f90	47
src/spectrum.f90	47
src/spline_akima.f	47
src/spline_akima_int.f	47
src/spline_cubic.f	47
src/spline_cubic_int.f	48
src/symforce.f90	48
src/symrzi.f90	48
src/tolower.f90	48
src/tomnsp.f90	48
src/totzsp.f90	48
src/tridslv.f90	49
src/vmec.f90	50
src/wrout.f90	50
src/data/fbal.f90	11
src/data/realspace.f90	11
src/data/stel_constants.f	12
src/data/stel_kinds.f	12
src/data/vforces.f90	13
src/data/vmec_dim.f90	13
src/data/vmec_input.f90	14
src/data/vmec_io.f90	15
src/data/vmec_main.f90	15
src/data/vmec_params.f90	18
src/data/vmec_persistent.f90	19
src/data/vmercier.f90	20
src/data/vparams.f90	20
src/data/vsvd0.f90	21
src/data/xstuff.f90	21
src/NESTOR/analysum.f90	31
src/NESTOR/analysum2.f90	31
src/NESTOR/analyt.f90	31
src/NESTOR/becoil.f90	31
src/NESTOR/belicu.f90	32
src/NESTOR/bextern.f90	32
src/NESTOR/fouri.f90	35
src/NESTOR/fourp.f90	36

src/NESTOR/ <a href="#">greenf.f90</a> . . . . .	36
src/NESTOR/ <b>nestor_main.f90</b> . . . . .	??
src/NESTOR/ <a href="#">precal.f90</a> . . . . .	36
src/NESTOR/ <a href="#">scalpot.f90</a> . . . . .	36
src/NESTOR/ <a href="#">surface.f90</a> . . . . .	36
src/NESTOR/ <a href="#">vacuum.f90</a> . . . . .	36
src/NESTOR/data/ <b>nestor_io.f90</b> . . . . .	??
src/NESTOR/data/ <a href="#">vac_persistent.f90</a> . . . . .	32
src/NESTOR/data/ <a href="#">vacmod.f90</a> . . . . .	32
src/NESTOR/data/ <a href="#">vacmod0.f90</a> . . . . .	35



## Chapter 3

# Data Type Documentation

### 3.1 read\_wout\_mod::read\_wout\_file Interface Reference

#### Public Member Functions

- subroutine **readw\_and\_open** (file\_or\_extension, ierr, iopen)

#### 3.1.1 Detailed Description

Definition at line 236 of file read\_wout\_mod.f.



## Chapter 4

# File Documentation

### 4.1 src/add\_fluxes.f90 File Reference

#### Functions/Subroutines

- subroutine **add\_fluxes** (overg, bsupu, bsupv)

### 4.2 src/alias.f90 File Reference

#### Functions/Subroutines

- subroutine **alias** (gcons, ztemp, gcs, gsc, gcc, gss)

### 4.3 src/allocate\_func3d.f90 File Reference

#### Functions/Subroutines

- subroutine **allocate\_func3d**

### 4.4 src/allocate\_ns.f90 File Reference

#### Functions/Subroutines

- subroutine **allocate\_ns** (linterp, neqs\_old)

### 4.5 src/allocate\_nunv.f90 File Reference

#### Functions/Subroutines

- subroutine **allocate\_nunv**

## 4.6 src/aspectratio.f90 File Reference

### Functions/Subroutines

- real(rprec) function **aspectratio** ()

## 4.7 src/bcovar.f90 File Reference

### Functions/Subroutines

- subroutine **bcovar** (lu, lv)

### 4.7.1 Function/Subroutine Documentation

#### 4.7.1.1 bcovar()

```
subroutine bcovar (
    real(rprec), dimension(nrzt,0:1), intent(inout) lu,
    real(rprec), dimension(nrzt,0:1), intent(inout) lv )
```

R12 from RP in force

Norm, unpreconditioned R,Z forces

Norm for preconditioned R,Z forces

Norm for unpreconditioned Lambda force

Definition at line 2 of file bcovar.f90.

Here is the caller graph for this function:



## 4.8 src/bextrema.f90 File Reference

### Functions/Subroutines

- subroutine **bextrema** (modb, bmin, bmax, nzeta, ntheta)



## 4.9 src/bss.f90 File Reference

### Functions/Subroutines

- subroutine **bss** (r12, rs, zs, ru12, zu12, bsubs, bsupu, bsupv, br, bphi, bz)

## 4.10 src/calc\_fbal.f90 File Reference

### Functions/Subroutines

- subroutine **calc\_fbal** (bsubu, bsubv)

## 4.11 src/convert.f90 File Reference

### Functions/Subroutines

- subroutine **convert** (rmnc, zmns, lmns, rmns, zmnc, lmnc, rzl\_array, js)

## 4.12 src/data/fbal.f90 File Reference

### 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**

## 4.13 src/data/realspace.f90 File Reference

### 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`

## 4.14 `src/data/stel_constants.f` File Reference

### 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`

## 4.15 `src/data/stel_kinds.f` File Reference

### 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`

## 4.16 src/data/vforces.f90 File Reference

### 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::bzmn**
- 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::bzmn\_e**
- real(rprec), dimension(:), pointer **vforces::bzmn\_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**

## 4.17 src/data/vmec\_dim.f90 File Reference

### 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::mnsize**
- integer **vmec\_dim::mnmax\_nyq**
- integer **vmec\_dim::ns**
- integer **vmec\_dim::ns1**
- integer **vmec\_dim::ns\_maxval**

## 4.18 src/data/vmec\_input.f90 File Reference

### 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**

## 4.19 src/data/vmec\_io.f90 File Reference

### 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**

## 4.20 src/data/vmec\_main.f90 File Reference

### 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::bucv`
- `real(rprec), dimension(:), allocatable vmec_main::bvco`
- `real(rprec), dimension(:), allocatable vmec_main::bdotgradv`
- `real(rprec), dimension(:), allocatable vmec_main::equif`
- `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::bucv`
- `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::bsub0**
- 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::lthreed**
- 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::niter**  
*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

## 4.21 src/data/vmec\_params.f90 File Reference

### 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*

## 4.22 src/data/vmec\_persistent.f90 File Reference

### 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::sinnavn`
- `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`

## 4.23 src/data/vmercier.f90 File Reference

### Variables

- `real(rprec), dimension(nsd) vmmercier::dshear`
- `real(rprec), dimension(nsd) vmmercier::dwell`
- `real(rprec), dimension(nsd) vmmercier::dcurr`
- `real(rprec), dimension(nsd) vmmercier::dmerc`
- `real(rprec), dimension(nsd) vmmercier::dgeod`

## 4.24 src/data/vparams.f90 File Reference

### 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

## 4.25 src/data/vsvd0.f90 File Reference

### Variables

- integer, parameter **vsvd0::nigroup** = 100  
*number of external current groups*

## 4.26 src/data/xstuff.f90 File Reference

### 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**

## 4.27 src/elongation.f90 File Reference

### Functions/Subroutines

- subroutine **elongation** (r1, z1, waist, height)

## 4.28 src/eqfor.f90 File Reference

### Functions/Subroutines

- subroutine **eqfor** (br, bz, bsubu, bsubv, tau, rzl\_array, ier\_flag)

## 4.29 src/eqsolve.f90 File Reference

### Functions/Subroutines

- subroutine **eqsolve** (ier\_flag)

## 4.30 src/evolve.f90 File Reference

### Functions/Subroutines

- subroutine **evolve** (time\_step, ier\_flag, liter\_flag)

## 4.31 src/fileout.f90 File Reference

### Functions/Subroutines

- subroutine **fileout** (ier\_flag)

## 4.32 src/fixaray.f90 File Reference

### Functions/Subroutines

- subroutine **fixaray**

## 4.33 src/flip\_theta.f90 File Reference

### Functions/Subroutines

- subroutine **flip\_theta** (rmn, zmn, lmn)

### 4.33.1 Function/Subroutine Documentation

#### 4.33.1.1 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 )
```

## Parameters

in, out	<i>lmn</i>	never used: can also flip lambda...
---------	------------	-------------------------------------

Definition at line 2 of file flip\_theta.f90.

## 4.34 src/forces.f90 File Reference

### Functions/Subroutines

- subroutine **forces**

## 4.35 src/free\_mem\_func3d.f90 File Reference

### Functions/Subroutines

- subroutine **free\_mem\_func3d**

## 4.36 src/free\_mem\_ns.f90 File Reference

### Functions/Subroutines

- subroutine **free\_mem\_ns**

## 4.37 src/free\_mem\_nunv.f90 File Reference

### Functions/Subroutines

- subroutine **free\_mem\_nunv**

## 4.38 src/freeb\_data.f90 File Reference

### Functions/Subroutines

- subroutine **freeb\_data** (rmnc, zmns, rmns, zmnc, bmodmn, bmodmn1)

## 4.39 src/fsym\_fft.f90 File Reference

### Functions/Subroutines

- subroutine **fext\_fft** (bout, bs\_s, bs\_a)
- subroutine **fsym\_fft** (bs, bu, bv, bs\_s, bu\_s, bv\_s, bs\_a, bu\_a, bv\_a)

## 4.40 src/fsym\_invfft.f90 File Reference

### Functions/Subroutines

- subroutine **fsym\_invfft** (bssu, bssv)

## 4.41 src/funct3d.f90 File Reference

### Functions/Subroutines

- subroutine **funct3d** (ier\_flag)

### 4.41.1 Function/Subroutine Documentation

#### 4.41.1.1 funct3d()

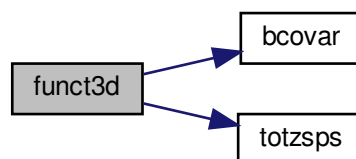
```
subroutine funct3d (
    integer, intent(inout) ier_flag )
```

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 2 of file funct3d.f90.

Here is the call graph for this function:



## 4.42 src/functions.f File Reference

### Functions/Subroutines

- real(rprec) function, public **functions::two\_power** (x, b)
- real(rprec) function, public **functions::two\_power\_gs** (x, b)
- logical function **functions::function\_test** ()

## 4.43 src/getbsubs.f90 File Reference

### Functions/Subroutines

- subroutine **getbsubs** (bsubsmn, frho, bsupu, bsupv, mmax, nmax, info)

## 4.44 src/getcurmid.f90 File Reference

### Functions/Subroutines

- subroutine **getcurmid** (curmid, izeta, gsqrt, r12)

## 4.45 src/getfsq.f90 File Reference

### Functions/Subroutines

- subroutine **getfsq** (gcr, gcx, gnormr, gnormz, gnorm, medge)

## 4.46 src/guess\_axis.f90 File Reference

### Functions/Subroutines

- subroutine **guess\_axis** (r1, z1, ru0, zu0)

## 4.47 src/heading.f90 File Reference

### Functions/Subroutines

- subroutine **heading** (extension)

## 4.48 src/initialize\_radial.f90 File Reference

### Functions/Subroutines

- subroutine **initialize\_radial** (nsval, ns\_old, delt0)

## 4.49 src/interp.f90 File Reference

### Functions/Subroutines

- subroutine **interp** (xnew, xold, scalxc, nsnew, nsold)

## 4.50 src/jacobian.f90 File Reference

### Functions/Subroutines

- subroutine **jacobian**

## 4.51 src/jxbforce.f90 File Reference

### Functions/Subroutines

- subroutine **jxbforce** (bsupu, bsupv, bsubu, bsubv, bsubsh, bsubsu, bsubsv, gsqrt, bsq, itheta, izeta, brho, ier\_flag)

## 4.52 src/lamcal.f90 File Reference

### Functions/Subroutines

- subroutine **lamcal** (overg, guu, guv, gvv)

## 4.53 src/line\_segment.f File Reference

### 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** ()

## 4.54 src/magnetic\_fluxes.f90 File Reference

### Functions/Subroutines

- real(rprec) function **torflux\_deriv** (x)
- real(rprec) function **polflux\_deriv** (x)
- real(rprec) function **torflux** (x)
- real(rprec) function **polflux** (x)

### 4.54.1 Function/Subroutine Documentation

#### 4.54.1.1 polflux()

```
real(rprec) function polflux (
    real(rprec), intent(in) x )
```

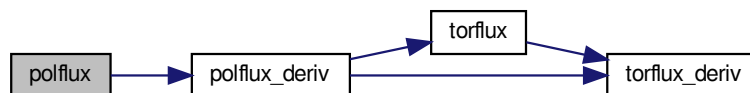


## Parameters

in	x	radial flux variable (=TOROIDAL FLUX ONLY IF APHI=1)
----	---	--

Definition at line 64 of file magnetic\_fluxes.f90.

Here is the call graph for this function:



## 4.54.1.2 polflux\_deriv()

```

real(rprec) function polflux_deriv (
    real(rprec), intent(in) x )

```

## Parameters

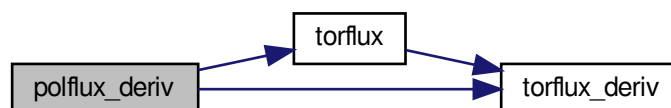
in	x	radial flux variable (=TOROIDAL FLUX ONLY IF APHI=1)
----	---	--

## Returns

`polflux_deriv == d(chi)/dx = iota(TF(x)) * torflux_deriv(x)`

Definition at line 21 of file magnetic\_fluxes.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.54.1.3 torflux()

```

real(rprec) function torflux (
    real(rprec), intent(in) x )
  
```

##### Parameters

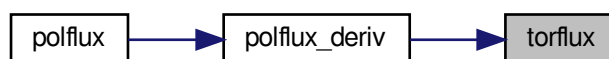
in	x	radial flux variable (=TOROIDAL FLUX ONLY IF APhi=1)
----	---	--

Definition at line 42 of file magnetic\_fluxes.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.54.1.4 torflux\_deriv()

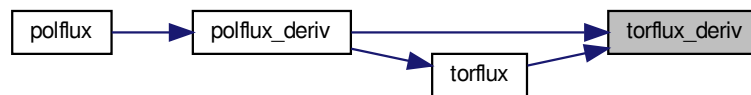
```
real(rprec) function torflux_deriv (
    real(rprec), intent(in) x )
```

##### Parameters

in	x	radial flux variable (=TOROIDAL FLUX ONLY IF APhi=1)
----	---	--

Definition at line 2 of file magnetic\_fluxes.f90.

Here is the caller graph for this function:



## 4.55 src/mercier.f90 File Reference

### Functions/Subroutines

- subroutine **mercier** (gsqrt, bsq, bdotj, iotas, wint, r1, rt, rz, zt, zz, bsubu, vp, phips, pres, ns, nznt)

## 4.56 src/mgrid\_mod.f File Reference

### 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::bvzac**
- 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**

## 4.57 src/NESTOR/analysum.f90 File Reference

### Functions/Subroutines

- subroutine **analysum** (grpmn, bvec, sl, tl, m, n, l, ivacskip, lasym, m\_map, n\_map, grpmn\_m\_map, grpmn\_n\_map)

## 4.58 src/NESTOR/analysum2.f90 File Reference

### Functions/Subroutines

- subroutine **analysum2** (grpmn, bvec, m, n, l, ivacskip, lasym, m\_map, n\_map, grpmn\_m\_map, grpmn\_n\_map)

## 4.59 src/NESTOR/analyt.f90 File Reference

### Functions/Subroutines

- subroutine **analyt** (grpmn, bvec, ivacskip, lasym, m\_map, n\_map, grpmn\_m\_map, grpmn\_n\_map)

## 4.60 src/NESTOR/becoil.f90 File Reference

### Functions/Subroutines

- subroutine **becoil** (rad, zee, brvac, bpvac, bzvac)

## 4.61 src/NESTOR/belicu.f90 File Reference

### Functions/Subroutines

- subroutine **belicu** (torcur, bx, by, bz, cos1, sin1, rp, zp)

## 4.62 src/NESTOR/bextern.f90 File Reference

### Functions/Subroutines

- subroutine **bextern** (plascur, wint)

## 4.63 src/NESTOR/data/vac\_persistent.f90 File Reference

### 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**

## 4.64 src/NESTOR/data/vacmod.f90 File Reference

### 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::bzv**
- 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::sqrta`
- `real(rprec), dimension(:), allocatable vacmod::sqrta`
- `real(rprec), dimension(:), allocatable vacmod::t1p2`
- `real(rprec), dimension(:), allocatable vacmod::t1p1`
- `real(rprec), dimension(:), allocatable vacmod::t1p`
- `real(rprec), dimension(:), allocatable vacmod::t1m2`
- `real(rprec), dimension(:), allocatable vacmod::t1m1`
- `real(rprec), dimension(:), allocatable vacmod::t1m`
- `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_t1p`
- `real(rprec), dimension(:, :), allocatable vacmod::all_t1m`
- `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`



## 4.65 src/NESTOR/data/vacmod0.f90 File Reference

### 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**

## 4.66 src/NESTOR/fouri.f90 File Reference

### Functions/Subroutines

- subroutine **fouri** (grpmn, gsource, amatrix, amatsq, bvec, wint, lasym)

### 4.66.1 Function/Subroutine Documentation

#### 4.66.1.1 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.

## 4.67 src/NESTOR/fourp.f90 File Reference

### Functions/Subroutines

- subroutine **fourp** (grpmn, grp)

## 4.68 src/NESTOR/greenf.f90 File Reference

### Functions/Subroutines

- subroutine **greenf** (delgr, delgrp, ip)

## 4.69 src/NESTOR/precal.f90 File Reference

### Functions/Subroutines

- subroutine **precal**

## 4.70 src/NESTOR/scalpot.f90 File Reference

### Functions/Subroutines

- subroutine **scalpot** (bvec, amatrix, wint, ivacskip, lasym, m\_map, n\_map)

## 4.71 src/NESTOR/surface.f90 File Reference

### Functions/Subroutines

- subroutine **surface** (rc, rs, zs, zc, xm, xn, mnmax, lasym, signgs)

## 4.72 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)

## 4.73 src/open\_output\_files.f90 File Reference

### Functions/Subroutines

- subroutine **open\_output\_files** (extension, lfirst)

## 4.74 src/parse\_extension.f File Reference

### Functions/Subroutines

- subroutine **parse\_extension** (file\_to\_parse, file\_or\_extension, Inc)

## 4.75 src/precondn.f90 File Reference

### Functions/Subroutines

- subroutine **precondn** (lu1, bsq, gsqr, r12, xs, xu12, xue, xuo, xodd, axm, axd, bxm, bxd, cx, eqfactor, trigmult)

## 4.76 src/printout.f90 File Reference

### Functions/Subroutines

- subroutine **printout** (i0, delt0, w0)

## 4.77 src/profil1d.f90 File Reference

### Functions/Subroutines

- subroutine **profil1d** (xc, xcdot, lreset)

## 4.78 src/profil3d.f90 File Reference

### Functions/Subroutines

- subroutine **profil3d** (rmn, zmn, lreset)

## 4.79 src/profile\_functions.f File Reference

### Functions/Subroutines

- real(rprec) function **pcurr** (xx)
- real(rprec) function **piota** (x)
- real(rprec) function **pmass** (xx)

## 4.80 src/read\_indata.f90 File Reference

### Functions/Subroutines

- subroutine **read\_indata** (in\_file, iunit, ier\_flag)

## 4.81 src/read\_wout\_mod.f File Reference

### Data Types

- interface [read\\_wout\\_mod::read\\_wout\\_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::loadrzi**

### 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** = 'signs'
- 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\_phip** = '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\_equif** = '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\_fsqt** = 'fsqt'
- 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: Poroidal 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** = 'bsubv 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\_phip** = '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 bsubv 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 bsubv 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::pref`
- `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**

## 4.82 src/readin.f90 File Reference

### Functions/Subroutines

- subroutine **readin** (input\_file, ier\_flag)

## 4.83 src/reset\_params.f90 File Reference

### Functions/Subroutines

- subroutine [reset\\_params](#)

### 4.83.1 Function/Subroutine Documentation

#### 4.83.1.1 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.

## 4.84 src/residue.f90 File Reference

### Functions/Subroutines

- subroutine **residue** (gcr, gcx, gcl)
- subroutine **constrain\_m1** (gcr, gcx)
- subroutine **scale\_m1** (gcr, gcx)

## 4.85 src/restart\_iter.f90 File Reference

### Functions/Subroutines

- subroutine **restart\_iter** (time\_step)

## 4.86 src/safe\_open\_mod.f File Reference

### Functions/Subroutines

- subroutine **safe\_open\_mod::safe\_open** (iunit, istat, filename, filestat, fileform, record\_in, access\_in, delim\_in)

## 4.87 src/scalfor.f90 File Reference

### Functions/Subroutines

- subroutine **scalfor** (gcx, axm, bxm, axd, bxd, cx, iflag)

## 4.88 src/solver.f90 File Reference

### Functions/Subroutines

- subroutine **solver** (amat, b, m, nrhs, info)

## 4.89 src/spectrum.f90 File Reference

### Functions/Subroutines

- subroutine **spectrum** (rmn, zmn)

## 4.90 src/spline\_akima.f File Reference

### Functions/Subroutines

- subroutine **spline\_akima** (x, y, xx, yy, npts, iflag)

## 4.91 src/spline\_akima\_int.f File Reference

### Functions/Subroutines

- subroutine **spline\_akima\_int** (x, y, xx, yy, npts, iflag)

## 4.92 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)

## 4.93 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)

## 4.94 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)

## 4.95 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)

## 4.96 src/tolower.f90 File Reference

### Functions/Subroutines

- subroutine **tolower** (string)

## 4.97 src/tomnsp.f90 File Reference

### Functions/Subroutines

- subroutine **tomnsp** (frzl\_array, armn, brmn, crmn, azmn, bzm, czmn, blmn, clmn, arcn, azcn)
- subroutine **tomnsps** (frzl\_array, armn, brmn, crmn, azmn, bzm, czmn, blmn, clmn, arcn, azcn)

## 4.98 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)

## 4.98.1 Function/Subroutine Documentation

### 4.98.1.1 totzsps()

```

subroutine totzsps (
    real(rprec), dimension(ns,0:ntor,0:mpoll,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.

Here is the caller graph for this function:



## 4.99 src/tridslv.f90 File Reference

### Functions/Subroutines

- subroutine **tridslv** (a, d, b, c, jmin, jmax, mnd1, ns, nrhs)

## 4.100 `src/vmec.f90` File Reference

### Functions/Subroutines

- program `vmec`

## 4.101 `src/wrout.f90` File Reference

### Functions/Subroutines

- subroutine `wrout` (`bsq`, `gsqrt`, `bsubu`, `bsubv`, `bsubs`, `bsupv`, `bsupu`, `rzl_array`, `gc_array`, `ier_flag`)



# Index

bcovar  
    bcovar.f90, 10  
bcovar.f90  
    bcovar, 10

flip\_theta  
    flip\_theta.f90, 22  
flip\_theta.f90  
    flip\_theta, 22

fouri  
    fouri.f90, 35  
fouri.f90  
    fouri, 35

funct3d  
    funct3d.f90, 24  
funct3d.f90  
    funct3d, 24

magnetic\_fluxes.f90  
    polflux, 26  
    polflux\_deriv, 27  
    torflux, 28  
    torflux\_deriv, 28

polflux  
    magnetic\_fluxes.f90, 26  
polflux\_deriv  
    magnetic\_fluxes.f90, 27

read\_wout\_mod::read\_wout\_file, 7  
reset\_params  
    reset\_params.f90, 46  
reset\_params.f90  
    reset\_params, 46

src/add\_fluxes.f90, 9  
src/alias.f90, 9  
src/allocate\_funct3d.f90, 9  
src/allocate\_ns.f90, 9  
src/allocate\_nunv.f90, 9  
src/aspectratio.f90, 10  
src/bcovar.f90, 10  
src/bextrema.f90, 10  
src/bss.f90, 11  
src/calc\_fbal.f90, 11  
src/convert.f90, 11  
src/data/fbal.f90, 11  
src/data/realspace.f90, 11  
src/data/stel\_constants.f, 12  
src/data/stel\_kinds.f, 12  
src/data/vforces.f90, 13  
src/data/vmec\_dim.f90, 13  
src/data/vmec\_input.f90, 14  
src/data/vmec\_io.f90, 15  
src/data/vmec\_main.f90, 15  
src/data/vmec\_params.f90, 18  
src/data/vmec\_persistent.f90, 19  
src/data/vmercier.f90, 20  
src/data/vparams.f90, 20  
src/data/vsvd0.f90, 21  
src/data/xstuff.f90, 21  
src/elongation.f90, 21  
src/eqfor.f90, 21  
src/eqsolve.f90, 22  
src/evolve.f90, 22  
src/fileout.f90, 22  
src/fixaray.f90, 22  
src/flip\_theta.f90, 22  
src/forces.f90, 23  
src/free\_mem\_funct3d.f90, 23  
src/free\_mem\_ns.f90, 23  
src/free\_mem\_nunv.f90, 23  
src/freeb\_data.f90, 23  
src/fsym\_fft.f90, 23  
src/fsym\_invfft.f90, 24  
src/funct3d.f90, 24  
src/functions.f, 24  
src/getbsubs.f90, 25  
src/getcurmid.f90, 25  
src/getfsq.f90, 25  
src/guess\_axis.f90, 25  
src/heading.f90, 25  
src/initialize\_radial.f90, 25  
src/interp.f90, 25  
src/jacobian.f90, 26  
src/jxbforce.f90, 26  
src/lamcal.f90, 26  
src/line\_segment.f, 26  
src/magnetic\_fluxes.f90, 26  
src/mercier.f90, 29  
src/mgrid\_mod.f, 29  
src/NESTOR/analysum.f90, 31  
src/NESTOR/analysum2.f90, 31  
src/NESTOR/analyt.f90, 31  
src/NESTOR/becoil.f90, 31  
src/NESTOR/belicu.f90, 32  
src/NESTOR/bextern.f90, 32  
src/NESTOR/data/vac\_persistent.f90, 32  
src/NESTOR/data/vacmod.f90, 32  
src/NESTOR/data/vacmod0.f90, 35

- src/NESTOR/fouri.f90, [35](#)
- src/NESTOR/fourp.f90, [36](#)
- src/NESTOR/greenf.f90, [36](#)
- src/NESTOR/precalf.f90, [36](#)
- src/NESTOR/scalpot.f90, [36](#)
- src/NESTOR/surface.f90, [36](#)
- src/NESTOR/vacuum.f90, [36](#)
- src/open\_output\_files.f90, [36](#)
- src/parse\_extension.f, [37](#)
- src/precondn.f90, [37](#)
- src/printout.f90, [37](#)
- src/profil1d.f90, [37](#)
- src/profil3d.f90, [37](#)
- src/profile\_functions.f, [37](#)
- src/read\_indata.f90, [38](#)
- src/read\_wout\_mod.f, [38](#)
- src/readin.f90, [46](#)
- src/reset\_params.f90, [46](#)
- src/residue.f90, [46](#)
- src/restart\_iter.f90, [46](#)
- src/safe\_open\_mod.f, [47](#)
- src/scalfor.f90, [47](#)
- src/solver.f90, [47](#)
- src/spectrum.f90, [47](#)
- src/spline\_akima.f, [47](#)
- src/spline\_akima\_int.f, [47](#)
- src/spline\_cubic.f, [47](#)
- src/spline\_cubic\_int.f, [48](#)
- src/symforce.f90, [48](#)
- src/symrzi.f90, [48](#)
- src/tolower.f90, [48](#)
- src/tomnsp.f90, [48](#)
- src/totzsp.f90, [48](#)
- src/tridslv.f90, [49](#)
- src/vmec.f90, [50](#)
- src/wrout.f90, [50](#)
  
- torflux
  - magnetic\_fluxes.f90, [28](#)
- torflux\_deriv
  - magnetic\_fluxes.f90, [28](#)
- totzsp.f90
  - totzsps, [49](#)
- totzsps
  - totzsp.f90, [49](#)