# FORTEC-3D Manual ver. 1.4

For FORTEC-3D ver 3.3 (open-source version)

2019/03/08 Shinsuke Satake
National Institute for Fusion Science, Japan

## General condition for the use of FORTEC-3D code

1. FORTEC-3D is a free software WITHOUT ANY WARRANTY. You can use, redistribute, and modify it under the term of the GNU General Public License, except for Mersenne-Twister random number generator part of the source code (module VAR_MT in Module-F3D_v3-3.f90).
2. Concerning Mersenne-Twister, follow the terms written in "license-DCMT.txt". In FORTEC-3D, only the random number generator subroutines (sgrnd and grnd), which have been tuned up by Satake, are contained.
3. Some pre- and post- process programs for FORTEC-3D uses GSPS library to make postscript figures. GSPS was originally developed in NIFS but is not maintained anymore. Users can omit the subroutines which depend on GSPS.
4. The Copyright of the original FORTEC-3D source code belongs to Shinsuke Satake (SS).
5. SS does not take any responsibility for any bugs in FORTEC-3D or any damage caused by the code.
6. Though it is opened to the public under the GPL license, SS never agrees with any activities to divert FORTEC-3D to harmful or martially purpose.
7. The users should state the name of FORTEC-3D when publishing research results obtained from the code, with proper citations.
8. Any question about the code, contact SS ( satake@nifs.ac.jp ) .
9. FORTEC-3D source code, manual, examples, and their bug-fix are available at https://github.com/satakeshinsuke/fortec-3d .

**FORTEC-3D :**
**Finite-ORbit-width Transport with Electric-field and Collisions in 3D configuration**

## Contents

# 1. How to compile the source code

FORTEC-3D consists of several Fortran90 source code files.
In the top directly (v3-3_final/), use “make” command to compile.
Depending on the environment of the computer, one needs to adjust the compiling options in Makefile.

**Requirements:**

- Fortran90 (or later) compiler, which is capable of compiling MPI & OpenMP hybrid parallel program.
- It is expected that the compiler can handle the mixed auto thread-parallel (by using “-parallel” option or something like that) and OpenMP explicit thread-parallel. It is assumed that compiler automatically parallelize the loops with simple structure and the array operations like “A(:,:)=B(:,:)*C(:,:)”.
- If your compiler cannot do that, please add OpenMP instructions into the source code where you can expect speed-up by parallelizing the loops.
- FORTEC-3D requires BLAS/LAPACK libraries: Subroutine DGESV.

**NOTE for computation efficiency :**

1. Subroutine W_AVE in wave2.f90 assumes that there are at least 8 OpenMP threads available to parallelize ( 4 threads in LINEARC in linear-c-src_ie.f90). If the user’s computer has smaller numbers of cores per a MPI rank, it is better to combine some of the “case (n)” sections in the do-loops which start from “do i_thread=0, N” in the subroutine so that the number of the cases in these do-loops is equal to or less than the number of parallel threads available.

2. In W_AVE , overlapping the MPI communication with calculation is implemented by using OpenMP threads. If the user’s computer environment is not suitable for this tuning, it is better not to do so.

3. In subroutines CALC_BFIELD and VISC_FLUX_ER, a fine-tuning for SIMD and pipelining for Fujitsu FX100 (Plasma simulator in NIFS) are implemented.
   Since these two subroutines spend most of the computation time, it is recommend that the user tune the parameter “nppl” in module MOD_TMP (in Module-F3D_v3-3.f90) to achieve the highest computation performance. Note that the suitable value of nppl may depend on the number of Fourier modes of magnetic field selected in each simulation run.

Example: FX100 → nppl=100 , Xeon phi KNL → nppl=2~10

4. Ver. 3-3 has some obsolete style output files (out*.07, 27, 21) to watch time evolution of the neoclassical fluxes and so on. For quick post-process (using AV-FLVC-v3_4), it is preferable to read the binary files out*.107 and out*.108 instead of text files out*.07 and 27 (explained in § 7.2). The miscellaneous data in out*.21 is seldom to be refereed to. Post-process to read out*.21 (NEWHATTEN) is contained in the FORTEC-3D package, but it is maintained anymore.
   out*.07, 27, 21 will be abolished in future.

## 2. Description of input files

The number after each file name(#xx) represents of the I/O unit number to read the file. In FORTEC-3D, it is assumed that the path to each file is sent to the program by environment variables when submitting a batch job, like "setenv fu18 field-data.test" (on NIFS Plasma simulator, written in csh script) or "export FORT18= field-data.test" (with Intel Fortran compiler, written in bash script).
FORTEC-3D is written under the policy that "open" command is not used inside the program except for the output files to continue the job "cont_00x_****" (see Sec. 6 ).

If FORTEC-3D fails to read the field-data (ex. incorrect nrcell number and so on), you need to convert the endian of the binary file by using run-time option of the fortran compiler you use. This problem may happen if READFLD ( § 5.1) and FORTEC-3D runs on different computers.

- field-data.*** (#18)
  - It contains the data of magnetic field, temperature and density profiles.
  - To make a field-data file, use "READFLD" ( § 5.1) as well as VMEC and BOOZ_XFORM codes contained in STELLOPT package (ver.8.50 or later is recommended).

- gm_table.*** (#19) (required in case ifgsrk==1)
  - Table data of radial flux $\Gamma_i, \Gamma_e, Q_i,$ and $Q_e$ in $(r, E_r)$ parameter space. It is generated from either GSRAKE, DKES, or any other neoclassical transport code.
  - The radial flux data is used in FORTEC-3D to solve the radial current and the time evolution of the radial electric field $\epsilon_0 \epsilon_\perp \frac{\partial}{\partial t} E_r = -e(Z_i \Gamma_i - \Gamma_e)$, where one of the ion or electron flux is solved by FORTEC-3D while the other is referred to the gm_table data.
  - To learn how to prepare the gm_table file, see § 5.2.

- seed-mt19937-1024-v03 (#97)
  - The coefficients for the Dynamic Creation of parallel Mersenne-twister pseudo-random number generator. (https://github.com/MersenneTwister-Lab/dcmt)
  - It contains the random number generator setup data up to 1024 MPI parallel run.

- Input file for FORTEC-3D parameters (input-f3d***.dat) (#05)
  - It contains several control parameters to determine the time step sizes, number of

marker, number of Fourier modes, etc.

- The meaning of each parameters are explained in § 3.1.
- How to set the time step size → see § 5.3.

- Initial radial electric field profile data (input-er.***) (#20)
  - Specifies the radial electric field profile.
  - Details are explained in § 3.2

- Ion mean parallel flow profile data (input-ub.***) (#94)
  - Give ion parallel flow $< U_{||i} B >$ profile to include the electron-ion friction force in the FORTEC-3D electron simulation. It is required to evaluate correct bootstrap current.
  - The $< U_{||i} B >$ profile is provided by FORTEC-3D ion simulation or any other neoclassical code.
  - Details are explained in § 3.3

- Magnetic perturbation spectrum data (#95)
  - Specifies the external magnetic perturbations superimposed on the VMEC equilibrium field.
  - Details are explained in § 3.4

# 3. Description of input parameters

## 3.1 input-f3d***.dat file

**FORTEC-3D reads the "input-f3d_***.dat" assigned to file I/O unit #05 which is specified in the batch script.**

**In the following list, the parameters with ☆ mark is the recommended one.**
**Important parameters are marked with underlines.**

**namelist/read1/**

dtau : time step size of collision operator (dt = dtau*Tau).
Here, Tau is the ion-ion or electron-electron collision time at rho=r/a=xhalf.

nss : fineness of orbit calculation time step (dtonrm = dt/nss)

ni : marker number per MPI rank (total = ni*nproc[# of MPIs])
NOTE: ni should be divisible by the parameter nppl (SIMD-tuning parameter).

xhalf : radial position on which normalization factors for n, T, collision time, etc. are evaluated.

kido = 0 : normal calculation ☆
= 1 : orbit check (collision and weight averaging are controlled by other parameter)
jist, jinm,jskp: particle indexes of which orbits are output to #99
(jist,jist+jskp,jist+2*jskp,.....,jist+(jinm-1)*jskp)
kdstep = if time step > kdstep, kido=0 (return to normal run)

kcoll = 0 : full calculation (orbit + collision) ☆
= 1 : collision calculation only (for collision operator test)
= -1 : no collision (orbit only)

ifpfm = 0 : neglect the field-particle collision operator (kcoll==-1 case, too)
= 1 : use the field-particle operator Pfm ☆

npltconsv : check the conservation properties of collision operator in each (r, theta, zeta) – cell every npltconsv steps.
= 0 : turn off ☆
Note : To read the result, use "chk-consv" program. ( § 7.2.5)

ngo : Control the continuous jobs
= 0 : run from the time step (istp)=1

>= 1 : continue from the previous (ngo-1) job (file; cont_nnn_****, where nnn is the order of the job and **** is the number of MPI ranks)
= -1 : run from istp=1, but read existing out.14 to skip some of the initializing routines. (out.14 can be used for other jobs using the same field-data file)

nstep = end of calculation time step.

nprestep = time step running without evolving ww and Erad. (recommend: always =0. This parapeter is used only for debugging purpose)
lasym : = .true. for up-down asymmetric case (like a single-null tokamak)
mmxcut( as) = use only the primal mmxcut+1 modes of magnetic field spectra ("_as" for up-down asymmetry components) out of all the mmx+1 modes contained in the field-data file.
= 0: uses all the mmx+1 modes contained in the field-data file.
(Note: mmx+1 is specified when the field-data is created)

ifelc : = 0 for ion , 1 for electron
ifeifric: = 1 to include ion mean flow profile in C_ei operator (require input file input-ub.*** in #94, see § 3.3)

**namelist/cellin/**

kzmax,kamax,krmax = total numbers of cells in (zeta,theta,rho) directions to be plotted in sub. PLOTWGT (plotting the information of the averaged weight field, df distribution, and marker distribution in 5D phase space)
kzc,kac,krc : cell positions (zeta,theta,rho) to be plotted (Max: 50 cells in each dimension)
If kzc(kac,krc)=-1, then all nzcell(nacell,nrcell) are to be plotted.
Note: Set the parameter ngraph (explained below) to activate PLOTWGT routine.
Plotting too much data will cost a lot of time and generate huge size of files #08, #09 and #30.
To read the outputs requires "dfplot_global/xdfplot" program ( § 7.2.4)
If you do not use this information, set kzmax=kamax=krmax=0.

**namelist/wavein/**

ntime = the time interval to call the weight-averaging routine W_AVE every 'ntime' steps
Note: W_AVE requires huge amount of ALLGATHER-type communication. To reduce the communication time, set ntime =2~5 ☆ instead of ntime=1. Too large ntime would

be harmful, however.

ngraph = output average weight field data etc. at the cells specified in “namelist/cellin” in every '|ngraph|*ntime' time steps. (note: not every ngraph time steps!)

ngraph> 0 : output weight fields, df , and f_M distribution averaged over ngraph times.

nghaph<0 : output immediate information of the above.

ngraph=0 : do not output the distribution function info. ☆

Note: ngraph = 100 or some large value to obtain smooth plots of weight functions. This function generates huge size output data. If you do not need to check delta-f function and weight field, it is better to turn it off.

ifave = switch of weight-averaging procedure ( 1 : on ☆ , 0 : off )

istave = averaging procedure starts from istave step (recommend: 1☆)

fwave = specify damping rate of weight spreading.

fwave=F means the variance of weight will exponentially damp in 1/F collision time. Larger F means rapider damping. (recommend: fwave=0.1~1.0) Too strong fwave results in artificial increase in effective collisionality.

ifwchg =0 : off , >0 : check the change in ww and wp and so on (output every ifwchg time steps to #17)) (How to “health check” the simulation run using the data : § 7.2.2 )

ffixp : This parameter is obsolete : Set always 0.

ifrcyc = switch of marker recycling 0: off, 1: do recycling☆

irctime= Do recycle procedure every irctime steps. (recommend 50 – 100)

Note: irctime must be dividible by ntime.

idead : Lifetime of markers. Markers which violate limiters ( for rhoi,ww/wp,wp,v/vth ) idead*nss counts in total are killed and respawn from xbnin(1)<= rhoi <= xbnout(1) (markers killed by the weight and velocity filters),
or xbnin(2)<=rhoi<=xbnout(2) (killed outside of the plasma by virtual limiter).

xkill : radial position of virtual limiter filter for markers escaping outside (usually > 1.1)

limwovp : limit of weight ww/wp (if limwovp=0 → no limit)

limwp : limit of weight wp (if limwp=0 → no limit)

limvth : limit of marker speed v/v_th (at marker position, limvth=0 → no limit)

rfade : ww and wp are multiplied by this value (0<rfade<=1.0) when a marker hits the filters. It moderates the effect of recycling and noise by large-weight markers.

Note: Too strong filters results in recycle of too many markers and unphysical results. However, Too week filter makes simulation result noisy. Always check #15 and

#16 if filter is too strong or not. Compare the simulation results with different strength of filters and confirm that the choice of the filters does not affect the simulation result. ( To check： § 7.2.2 )

iffixnt ：Source-Sink term to reduce dn and dnT (0: off, 1: on☆)
ffixnt ：Strength of the adaptive source-sink term (0.2~1.0)
(ffixnt=F means that dn and dnT damps at the time scale of 1/F collision time)

Note: Adaptive Source-sink term is useful if FORTEC-3D cannot reach a quasi-steady state solution after several collision times. However, this function should be used with care, since it may affect the neoclassical fluxes obtained.

**namelist/efldin/**

ifref = 0 ： without radial electric field time evolution
= 1 ： radial electric field evolution with dtonrm=dt(coll.)/nss
inief = 0 ： initial E_r=0
=+-i： initial E_r= (i/10)*(analytic force-balance solution in tokamak when $\langle U_{\parallel} B \rangle$=0 )
=-1000: read E_r - data (on full mesh) from #20 (input-er***.dat, § 3.2 )
=-1500: read E_r - data (on half mesh) from #20 ☆
=-2000: give linear (iertype=1) or parabola (2), qubic (3) profile
erfit,erfit_max,erpos,erpos_max : specify the shape of E_r(rho)
Note : if inief==-1500, E_r profile is imported from #20 even for ngo>=1 continued job.
emag : magnification factor of E_r [E_r (rho)=emag * input profile data file]

ifgsrk = 1 : read gm_table file from #18.
= 0 : no Gamma_e input data ☆
Note: This option is usually used if you want to solve time evolution of E_r by solving ion species by FORTEC-3D and use table data of particle flux of electrons from GSRAKE or DKES.

iffiter0 = 0： Time evolution of Erad0(1) is solved normally (old FORTEC-3D).
= 1： Erad0(1) is fitted from Erad(2) (New scheme since 2013/9) ☆
= 2： epsilon_perp near the magnetic axis are artificially magnified to stabilize Er oscillation
Note: When you run an Er time-evolution simulation, iffiter0=1 or =2 scheme are more numerically stable around the magnetic axis, but not physically exact.

ifntm,ifnpm = 0 : Do not calculate toroidal and poloidal viscosity
(ifntm_as, ifnpm_as : asymmetry part)
>= 1 : specify the number of modes TVC and PVC are solved (<= max # of modes)
-1 : solve all the modes
(determined by the number of Bmn-Fourier modes used in a simulation)

Note : evaluation of NTV and NPV adds considerable computation time especially if there are many toroidal and poloidal Fourier modes exist.

**namelist/fun2in/**

iffun : switch to select the initial marker distribution in radial meshes.
= 1 : marker density in each radial mesh is proportional to (n*dV)
= 2,3: marker density is proportional to FUN2(x) ☆
FNA,FNB,FNC : coefficients for FUN2(x)

ifdist : 0 markers are distributed uniform in (theta,zeta)
: 1 markers are distributed uniform in volume (on each flux surface) ☆

Note: Radial population of the initial markers is proportional to
FUN2(x=r/a) = fna*x**2+fnb*x+fnc (iffun=2) , or
= fna*x**3+fnb*x**2+fnc*x+1.0 (iffun=3).
Specify proper coefficients according to the plasma density profile so that the initial weight wp has smooth profile in the radial direction and small variance ( Init. wp can be seen in #41). However, most of FORTEC-3D runs are done by simply setting iffun=2, FNA=FNB=0, FNC=1 (radially uniform initial marker distribution).

**namelist/magf/**

bmag : magnification factor for B-field
cmag : magnitication factor for collisionality
dmag : magnitication factor for density
ifgme =1 : Gamma-table is assumed to be proportional to cmag/bmag**2
=0 : Use Gamme-table as it is. ☆

**namelist/dblist/**

nbmn_c,_s : number of Fourier modes for magnetic perturbation (0: none)
If nbmn_c, _s =-1, magnetic perturbation profile is given by a data file (#95, see § 3.4)
dbmn_c,_s (i) : amplitude of the i-th MP component (relative to Bax)

cmdb,cndb (i) : (m,n) mode numbers of the i-th MP cos components)
smdb,sndb (i) : (m,n) mode numbers of the i-th MP (sin components)
ldb_c,_s (i) : ldb> 0 → dbmn radial profile (prop. to rho**ldb) [ldb is real number]
ldb<=0 → dbmn(rho) according to the function form specified in sub. READ_BFIELD_TABLE (exponential prof. centered at rho=0.5).

NOTE: The magnetic perturbation modes given by using DBLIST are inserted to the 3,4,5,…, nbmn_c(_s)+2 position of the Bmn components.
For example, if mmxcut+1=6 and the original equilibrium field data contains the following (m,n) spectra, and then if 2 components (3,20) (4,20) are added, the Fourier spectra used in the simulation becomes as follows :

| #index | orginal data (m,n) | after inserting MP modes | |
|---|---|---|---|
| 1 | 0,0 | 0,0 | ! first two modes are always 0,0 and 1,0 |
| 2 | 1,0 | 1,0 | |
| 3 | 2,10 | 3,20 | !inserted |
| 4 | 2,0 | 4,20 | !inserted |
| 5 | 1,10 | 1,0 | !original 2nd mode |
| 6 | 3,10 | 2,10 | !original 3rd mode |

## 3.2 input-er***.dat file

It gives FORTEC-3D the initial E_r profiles either on nercell full-meshes or half-meshes, where nercell is the number of radial mesh sizes specified in the field-data file.
full-mesh [half-mesh] means rho= i/nercell [ =(i-0.5)/nercell], (i=1,2,…,nercell).
The input-er file should be as follows:
(nercell=50 half-mesh case)

```
#   rho      E_r [V/m]    ← the first line is comment.
0.0100     0.040
0.0300     0.250
0.0500     1.450
…          …
0.9700    32.45
0.9900    43.45    (for full-mesh data, rho becomes 0.020,0.040,,,0.980,1.000)
```

If the **"inief"** parameter in the "input-***.dat" file is **-1000**, FORTEC-3D recognize the input-er dat is full-mesh data, and **-1500** means that the input data is on the half-meshes.
If the unit of E_r in the input-er.*** file is not [V/m], you can adjust it by the multiplying parameter "emag".

Note: In case of inief=-1500, E_r profile is reset to the profile given by the input file even if it is a

continuous job (ngo>=1).

## 3.3 Parallel flow profile data

For the electron simulation with parallel friction $F_{\parallel ei}$, you can give the finite ion background flux-surface average parallel mean flow profile $< U_{\parallel i} B > (\rho)$ [in Tm/s unit] from the file #94 (switch : ifeifric==1)

The data format is as follows:

```
#   rho     <UB>     ! first line is header
0.005        123.00
0.015        456.78
.....        .....
(total nercell radial points, on half-mesh)
```

## 3.4 Magnetic perturbation profile data

To specify the magnetic perturbation profile explicitly by a data file (nbmn_c(_s)==-1), prepare the data file in the following format and pass it to FORTEC-3D as #95 input file.

```
Line 1     nmptc,nrptc     ! number of modes and radial meshes   (for cos-components)
Line 2~   m(1)   n(1)   m(2)   n(2)   …. m(nmptc)   n(mnptc) ! (m,n) mode number list
(one blank line)
Line 4~   rdum1 rdum2   rho(1)   dB(1:nmptc,1)
          rdum1 rdum2   rho(2)   dB(1:nmptc,2)
               …….
          rdum1 rdum2   rho(nrptc)   dB(1:nmptc,nrptc)
(if sine-components are given, repeat the same data after the cosine-components)
```

Here, rdum1 and 2 are dummy real numbers (not used in FORTEC-3D)

The radial mesh positions (rho(1:nrptc)) can be given arbitrary. The dB radial profile is automatically spline-fitted in FORTEC-3D.

Then, all dB components are multiplied by dBmn_c(1) ( dBmn_s(1) for sine). Use this multiplication factors so that the given magnetic perturbation amplitudes be in Tesla unit.

(Note : This is different from that case nbmn_c,s > 0 , where the input dBmn_c,_s means the relative amplitude of magnetic perturbation, dB(m,n)/B_ax . )

You can check the dB profile by looking out.45.

# 4. Description of each subroutine

- ## Initialization
  - READ_BFIELD_TABLE
  - READ_VOL_TABLE
  - READ_GAMMA_TABLE

    Read field-data and gamma-table
  - SET_PARA_RANGE : Set parameters for MPI work-share and communications
  - TBLERR_ie : Prepare data-table for collision operator
  - RNxCALC : evaluate cell volume*density in (r,theta,zeta)-cells.
  - MARKDIST , INIT_WP : initial marker loading
  - RW_JOB_DATA : read and write continue file
  - ALLOC_VAR : allocate arrays decrared in modules
  - DFPPRE : preparation for PLOTWGT and DFPLOT

- ## Main loop
  - ORBIT
    - ✧ VISC_FLUX_ER : Evaluate transport and viscosity, time evolution of Er
    - ✧ RUNGE_KUTTA : Time integral of guiding-center equations of motions
      - ✧ CALCRK : 4th-order RK method
      - ✧ RERK4_NEW : RK routine for markers very close to the magnetic axis
    - ✧ NEWPOS : update marker cell position
    - ✧ FILTERS_WP, _VR : filters
  - RECYCLE_MARKER : marker recycling
  - LINEARC : collision term
    - ✧ CEI : e-i collision (pitch-angle scattering + F_ei friction)
    - ✧ CTP : Test-particle operator (like-species)
    - ✧ PFm_new : Field-particle operator (like-species) and Source-Sink term
  - W_AVE : Weight averaging
    - ✧ PLOTWGT, DFPLOT : export data of weight field and distribution function
  - BAD_LOG : export filter and recycling information
  - WRITE_ERGM, _QVP : export data of flux and viscosity

- ## Others
  - CALC_BFIELD : evaluate magnetic field, current at each marker's position
  - CALC_EFIELD : evaluate radial electric field at each marker's position
  - CALC_BACK_NT : evaluate n and T at each marker's position

- CHK_WSUM : make log data to “health check” the simulation run
- sgrnd, grnd : Mersenne-Twister random number generator
- ERAD_INTPL : update E_r interpolation table
- gauss2 : Make Gaussian distribution
- fit_x3, spline_fit2 : Make spline-fitting table
- val_intp : evaluate value using the spline table
- Calc_flux : evaluate several flux-surface quantities (diagnostics purpose, obsolete output style)
- FUN1, FUN2 : functions used to give initial radial distribution of markers

# 5. How to set up the input files

## 5.1 Creating field-data file

**Requires VMEC, BOOZ_XFORM, and READFLD programs.**

VMEC : Solve MHD equilibrium of 3D torus.
"wout" file (ver.8.50 or later is recommended), either .txt or .nc (NETCDF) format is acceptable for Booz_xform.

BOOZ_XFORM : Read wout file and transform the equilibrium data to Boozer coordinates.
Output: boozmn file

READFLD (written by Satake) : Translator from boozmn file to the field-data file for FORTEC-3D. It also includes the plasma density and temperature profiles. Input data for GSRAKE can be created from READFLD, too.

Note : Presently READFLD uses a plotting library GSPS developed in NIFS. If a user does not have GSPS or does not require the plotting of magnetic field spectrum, flux surface shape, etc. , please turn off the GSPS subroutine calls (remove "-DGSPS" flag in Makefile).

READFLD provides the output files which contains the information plotted by the library so that users can check the data by your own way.

**For more details, read "vmec-boozxform_readfield_eng.pdf".**

## 5.2 Creating gm_table file

There are two ways to make the "gm_table" data file, which has the table data of radial neoclassical particle and energy fluxes, $\Gamma_{i,e}$ and $Q_{i,e}$, $\langle U_{\parallel} B \rangle_{i,e}$ in the parameter space $(\rho, E_r)$ ( $Q_{i,e}$ and $\langle U_{\parallel} B \rangle_{i,e}$ are optional).

**The $\Gamma_{i,e}$-table is required to run a $E_r$ time-evolution simulation of FORTEC-3D.**

1 : Use **GSRAKE code** ( for LHD) (originally written by Massberg and Beidler, IPP)
Satake's modified version runs GSRAKE in parallel, and the gm_table is generated.

2 : Prepare $\Gamma_{i,e}$ ( and $Q_{i,e}$ and $\langle U_{\parallel} B \rangle_{i,e}$) table data by any other codes such as **DKES, DKES/PENTA, DGN/LHD,** and transform the result to generate gm_table file using **READ-DKES-TABLE** code (written by Satake and Velasco). Only single ion species case is allowed.

The user should prepare the radial flux table data in the following format.

---

```
# number of (rho, E_r) grids
22 200        ! ← data size of the flux table
#   s=rho^2        E_r [V/m]     Gamma_e[m/s]/n_e Gamma_i[m/s]/n_i   Q_e[m/s]/nT_e Q_i[m/s]/nT_i   vpar_e[m/s] vpar_i[m/s]
   3.00000E-03 -2.97500E+04   9.21389E-02   2.79069E-02              2.43830E-01    1.57811E-02    2.49020E+03    5.19121E+04
   3.00000E-03 -2.95000E+04   9.12998E-02   2.68211E-02              2.42044E-01    1.27165E-02    2.03677E+03    5.08905E+04
   3.00000E-03 -2.92500E+04   9.04626E-02   2.57237E-02              2.40260E-01    9.56932E-03    1.58645E+03    4.98731E+04
   3.00000E-03 -2.90000E+04   8.96275E-02   2.46133E-02              2.38478E-01    6.33305E-03    1.13897E+03    4.88597E+04
   3.00000E-03 -2.87500E+04   8.87944E-02   2.34886E-02              2.36698E-01    3.00091E-03    6.94629E+02    4.78506E+04
.....
   9.50000E-01   1.92500E+04 -2.02861E-01 -1.91317E-01             -8.18242E-01   -1.46355E-02   -2.32833E+04   -3.10866E+04
   9.50000E-01   1.95000E+04 -2.04910E-01 -1.96058E-01             -8.25989E-01   -2.32960E-02   -2.35847E+04   -3.14192E+04
   9.50000E-01   1.97500E+04 -2.06923E-01 -2.00770E-01             -8.33569E-01   -3.18548E-02   -2.38863E+04   -3.17518E+04
   9.50000E-01   2.00000E+04 -2.08901E-01 -2.05451E-01             -8.40982E-01   -4.03157E-02   -2.41882E+04   -3.20846E+04
```

---

Here, one can use either Γ or Γ/n for particle flux, Q or Q/T for energy flux. The radial coordinate can be either rho or s( in VMEC, s=rho^2). The switches for these choices are ifrho,ifginp,ifqinp, ifuinp (See the comments in the source code).

The READ_DKES_TABLE code then inter(extra-)polate the input table to the range rho=[0:1]. The range of $E_r$ is the same as the input table. Then, gm_table file, of which mesh size is (nercell, meshEfld), is generated.

The Γ(rho, $E_r$)-table of the input data can be unequal spacing, but the minimum and maximum values of $E_r$ on every flux surface should be the same.

The way to interpolate Γ(rho, $E_r$)-table is bi-Lagrange spline (order of iftbl). Recommendation is iftbl=1 (simple linear fitting).

Note: READ_DKES_TABLE code requires the field-data file for the same configuration.

## 5.3 Setting up the time step size before running FORTEC-3D

The "Chkdtau/dtauchk-v33.f90" program reads the field_data.*** file and input***.dat files and tells you the information about the time step size.

For good resolution of orbit calculations, it is recommended that $\tau_{\mathrm{transit}}/\Delta t_{\mathrm{orbit}}$>100 where $\tau_{\mathrm{transit}} = qR/(Nv_{th})$ (N is the number of helical pitch. for tokamak, N=1), while $\Delta t_{\mathrm{orbit}} = \tau_{ii}$ (or $\tau_{ee}$) $* dtau/nss$.
Also, if one run a simulation with time evolution of E_r, it is recommended that $\tau_{\mathrm{GAM}}/\Delta t_{\mathrm{col}}$>100, where $\tau_{\mathrm{GAM}}$ is inverse of GAM frequency (in circular tokamak) and $\Delta t_{\mathrm{col}} = \tau_{ii} * dtau$.
The radial profiles of these values appear in "chkdtau.out" (t_tr/dto and t_gam/dtc).

(plot-chkdtau.gpl is a sample to make figures from the output using Gnuplot)

In the shell script "chkdtau-v33.sh" to run the code, one needs to specify the numbers of mail helical modes (for example, (ihm,ihn)=(2,10) for LHD, =(1,5) for W7-X, etc.)
For axisymmetric case, set (ihm,ihn)=(0,0).
Using this program, you can also check the n and T profile and normalized collision frequency ($\nu^*_{\mathrm{banana}}, \nu^*_{\mathrm{h}}, \nu^*_{PS}$) of the kinetic profiles contained in a field-data file.
Switching the particle species (electron/ion) is done by the parameter "ifelc" (1 / 0) in the FORTEC-3D input file.

Note: For tokamak with RMP or toroidal ripples:

As "N" in evaluating $\tau_{\mathrm{transit}}$ is equal to ndivz in the field-data file, it is either 1 (for full-torus case) or 18 (if 1/18 section of torus is treated), for example.

If N=18, $\tau_{\mathrm{transit}}/\Delta t_{\mathrm{orbit}}$>100 is too severe condition to define the time step size. So, in case of tokamaks, set dtau and nss so that $\tau_{\mathrm{transit}} * \mathrm{N}/\Delta t_{\mathrm{orbit}}$>200~300.

In "Chkdtau/Sample", there are samples for LHD electron and ion cases. Use "plot-chkcol.gpl" to plot figures from the output files.

# 6. How to run FORTEC-3D

See the example case in “SAMPLE/ion(elc)-1st****” ( § 8) . You need to put field-data.****, input-f3d-v3-3***.dat, and optionally input-er.****, input-ub.****, and gm_table.**** files, to proper working directory specified in gof3d-v3-3.csh.

It is assumed that a single simulation starts with ‘ set ID=”0” ‘ , and increase the number according to “ngo” in input-f3d-v3-3***.dat when you continue a run.

- To run a long simulation until neoclassical fluxes converge, you need to continue several jobs. For example, if you continue 20000 steps x 3 times , do as follows:
  1st run : set ID=”0”, ngo=0, nstep=20000
  2nd run : set ID=”1”, ngo=1, nstep=40000
  3rd run : set ID=”2”, ngo=2, nstep=60000
  The “ngo=x” jobs reads “cont_00(x-1)_****” file to continue the previous job.
- Usually, supercomputer systems set a time limit to run a single job. Even if FORTEC-3D cannot reach to nstep before the time limt, it outputs cont_*** and out*** files at every 1000 steps check point. You can restart from the last check point by reading the cont_*** file. (The last check point timing is written in out*.06, “*** check point *** istp = xxxx” .)

The total population of simulation marker is “ ni * (number of MPIs) “. The best distribution of ni and MPI depends on your system. Usually ni=100000 ~ 500000 is used. To obtain a statistically good (i. e., less noisy) result, it is recommended that the total marker population is 5000 x nacell x nrcell x nzcell or more.

# 7. How to see the results

## 7.1 : Description of output files

**(Here, X indicates the ID of continuous jobs. “bin” means binary file.)**

outX.06 : General information of the job. User can see the progress of job during it is running by watching this.

outX.07 : Radial profiles of E_r and Gamma_i,e (radial flux evaluated from radial drift velocity v_dr) at each time steps.
(ifntv≠0 or ifnpv≠0) Gam_tv, Gam_pv are fluxes proportional to NTV and NPV (see Satake, PPCF 2012), and Gam_vnc is the radial flux evaluated from NTV and NPV. If a calculation has enough accuracy, Gamma_i(or e) from v_dr and Gam_vnc should coincide with.

outX.08, outX.09, outX.30 (bin): Profiles of delta-f distribution function, average weights, and interpolated weight fields, respectively.

To read these files requires a post-process code DFPLOT ( § *.*).

out.12, out.13: Radial profiles of density, temperature etc. These two files are created only when ngo=0 or -1.

out.14 (bin) : Data of volume-integrated number of (plasma) particles in (r,theta,zeta) cells. This file is created only when ngo=0. In the continue jobs ngo=1,2,…, FORTEC-3D reads the existing out.14.

If you set ngo=-1, then FORTEC-3D reads the existing out.14 file and restarts from istp=1.

outX.15 : Log file of markers killed by the outside virtual limiter (n_all1), or filters (n_all2). n_max1 and n_max2 are maximum numbers of the killed markers among the MPI processes.

outX.16 : Precise log of filters (w/p, p, v/vth), and

psi<0 : number of markers (nss times sum) which are too close to the magnetic axis (rho<rho_na) and therefore the guiding-center motions of them are solved using a different subroutine (sub. RERK4_NEW).

v<prp0: number of markers of which v_perp became negative after test-particle collision operation.

outX.17: Check sum of marker weights ww and wp.

From ver.3-3, this output is created if ifwchg>0 (every ifwchg steps).

It contains detailed information about the change in ww and wp in each subroutines.

outX.21 (bin): Log of flux, flow, E_r etc. generated from sub. Calc_flux.

To read this file requires “NEWHATTEN” code.

(This function will be abolished in future, and completely replaced by the binary outputs #107 and #108)

outX.22 (bin) : Log file for Linearized collision operator and adaptive source-sink term.

To read this file requires “CHKCONSV” code ( § *.*).

outX.27 : Radial profiles of energy flux Q_i (or Q_e), parallel flow, flux-surface averaged density and pressure perturbation dn/(dn0), dnT/(dnT0) at each time steps.

outX.67 and outX.68 (bin): Radial profiles of neoclassical toroidal and poloidal viscosity.

NTV and NPV are decomposed into (m,n) modes (see Satake PPCF 2012).
In case of up-down asymmetry: outX.63 and outX.64 are also produced.

Note: from ver.1.1, normalization of these files has been changed. In the new FORTEC-3D, the values in #67 and #68 are in [N/m^2], while the older versions are in dimensionless quantity. (See also the usage of "AV-FLVC" code ( § *.*))

outX.107, outX.108 (bin): From v3-2a~

New format of Er, Gamma, Q, U*B, dn, dnT evolution data.
These files are intended to reduce the time to read outX.07 and outX.27 text-style files in order to speed up the post-process AV-FLVC.

Others (not important)

out.41, out.42, out.43 : Initial marker distribution and initial marker weight information.
out.45 : profile of external magnetic perturbation (if it is given)
outX.46 : Check log for the weight-averaging
outX.98 : Some parameters used in post-process programs.

## 7.2 : Post-process programs

Examples of the post process are found in “FORTEC-3D_v3-3_final/SAMPLE” directory.

### 7.2.1 : AV-FLVC/go-avflvc-v3

This program is used to take time average of the values in #07,#27,#67, #68 (+ #63 and #64 for up-down asymmetry case) files, which are separated if you run a series of continuous jobs. #107, #108 are binary format of #07 and #27.

Note :

1. This program can be run either serial or MPI parallel (4 processes are required). The parallel part is only when the output files are read. (for quick reading many data files). User can choose serial or parallel execution when compiling. Set the “-Cpp -DMPI” flag (-Cpp is preprocessor option in your compiler) in the Makefile for MPI-parallel version.
2. If you always read the binary output #107 and #108, non-MPI version will run fast enough.
3. In the same directory, source code for “go-findroot-v3” is included ( § 7.2.6). The Makefile also compiles “find-root-v3.f90”. Some of the subroutines are shared between two codes. “find-root-v3.f90” requires LAPACK library to compile.

The averaging set-up parameters are given in “input-av-v34.txt”. You can evaluate several ways of average.

**&param**

ntmp : step size of short-term averaging to see the evolution ( outputs to ***_tmp.dat”)

nstart , nend : timing for long-term average ( outputs to ***_ave.dat )

nercell : number of radial meshes ( same as in the FORTEC-3D runs )

scale : multiplication factor for radial flux. Usually =1.0.

ngost, ngoed : start and end job ID to be read in the post-process.

Ex: ngost =2, ngoed=4 → read out{2,3,4}.07 etc.

Note: the {nstart, nend} range should be contained in the jobs specified by ngost and ngoed

ifgsrk : =0 off, =1 on Read the gm_table and compare Gamma b/w FORTEC-3D and GSRAKE (or any other local neoclassical flux table)

lasym : .true. if you need to see the up-down asymmetric components of viscosity

ntv, mpv : numbers of toroidal and poloidal modes of viscosity to be written in the output files tvc_tmp.dat etc. ( ntv_s, mpv_s : up-down asymmetry parts )

0 : off, >0 : on, -1 : all contained in #67, #68 etc.

Note : If there are more than 50 modes, output is truncated and the sum of the truncated modes are written.

ifbin07 : =1 to read #107 and #108 (from FORTEC-3D ver. 3-3 ) instead of #07 and #27. The binary files are faster to be read.

**&norm**

sv0 : = 1.0 ( for compatibility with old style of viscosity output files)

lsmooth : .true. if you need smooth-fitting curves for Gamma, Q, UB, NTV, and NPV (outputs to ***_fit.dat )

mfit : order of the polynomials for the fitting curves (3<=mfit<nercell)

nrfit : fitted data are evaluated on nrfit+1 full-mesh grids including magnetic axis. ( convenient if one needs to fit the FORTEC-3D flux data to the different number of radial mesh points )

The ***_fit.dat files contains both nercell-mesh fitting data and nrfit-mesh data.

**&vol**

nsec : number of radial sections in which integral of viscosity torque is evaluated

nerin(1:nsec) ~ nerout(1:nsec) : the radial mesh positions of each volume

If this function is not required , set “nsec=1” and “nerin(1)=0”

**&stpin**

nstep = 0, … ,…., : list the end time steps of each jobs (up to ngoed job )

NOTE: the 1st item should always be “0”. Even if ngost>1, list up all the end steps.

To plot the data, please use “dev-ergm.gpl”, “prof-ergm.gpl” Gnuplot script files for example.

### 7.2.2 : gather17.sh, chk17.gpl

This script makes log plots of quick “health check” for FORTEC-3D simulation run whether the simulation runs correctly. Timing of writing data to out*.17 is controlled by the input parameter “ifwchg”.

**How to use:**

1. Assume two simulation data files are /work/case1, /work/case2 .
2. Put gather17.sh, chk17.gpl to /work/ .
3. Edit gather17.sh. Write the directory name list into “foreach DIRL ( )”
   Ex. : foreach DIRL (case1 case2)

Edit chk17.gpl. “totcel= ####” The #### number appears at the header of out0.46 .

4. Execute gather17.sh. It generates chk-f3d.ps under /work/caseX/ directory.
5. The meaning of each figure is as follows:

- page 1: Change in weight w (ww). “Sum” is sum. “RK, wave, filter, rcycl, coll, SS” represents the breakdown of the cause of change in ww (Runge-Kutta, weight-averaging, filter, recycle, collision, source/sink term , respectively).
- page 2 : Change in weight p (wp). The items are the same as in page 1.
  - ✧ The sum of ww should be << sum of wp (page 2) so that the delta-f ordering df/f_M << 1 is satisfied. The sum of wp should be almost unchanged during a simulation run, otherwise something is wrong [possible cause : time step size is not properly set, number of simulation markers (=ni*MPI) is too small, S/S or filter term to strong, too many simulation markers lost from the edge].
  - ✧ Collision and weight-averaging basically do not change the sum of ww. These items should be << 1 . In case of electrons with e-i friction force ( if ifeifric=1) , collision term does not completely conserve the particle number (=sum of ww), but still the change should be much smaller than the other causes.
  - ✧ Recycle should be zero for ww. When recycling, some recycled markers cannot determine its initial wp in the weight-averaging routine (if there is only a few markers exist near the phase space point where the marker respawns). In such a case, average wp at (rho, t=0) is assigned. This wp change is counted in the item “rcyc” of wp, and should be << Sum (wp).
  - ✧ Runge-Kutta part, or the weight change according to the guiding-center motion, should always satisfy ww+wp=0.
  - ✧ Filter represents the weight change caused by filters ( ww/wp > limwovp, wp > limwp, v/v_th>limvth, and lost particles from the boundary, rho > xkill). This sometimes becomes the main cause of weight change, but should be small compared to the sums of ww and wp.
  - ✧ Source/Sink term part in ww (controlled by ffixnt) will balance with RK and Filter terms in the quasi-steady. Confirm that SS(ww) << Sum (ww, wp).
  - ✧ Collision and SS do not affect wp at all.
- page 3 and 4 : ww and wp of dead markers by filters : These are just for debugging purpose. All items must be << 1 . Can be omitted from the plottings.

- page 5 and 6 : Log of W_AVE subroutine (weight averaging) [data = out*.46]
  - ✧ Total = total number of bins in the 5D phase-space (=totcel).
  - ✧ passed : weight averaging is carried out correctly.
  - ✧ nminwv : not enough markers (<6) in a bin to evaluate the averaged weight field.
  - ✧ Pedge<0 : The averaged wp field becomes <0 in the bin and therefore skipped.

- ✧ nomark : There is no marker in a bin.
- ✧ only-new : A marker respawned, but there is no existing ones to evaluate average weight field.
- ✧ cut-ww, cut-wp : Average-field is evaluated, but it does not conserves ww and wp before and after the averaging, so skipped.
- ✧ dW_ave>>0 : Average-ww field, which conserves the sum of the weight, is evaluated , but it is too much different from the simple arithmetic average ww. Then skipped.

Basically , items in page 6 are very few. It is preferable situation that the "passed" item is close to (at least more than 1/10 of ) the "Total" in page 5.

If marker population is very low compared to the total number of the bins, "nminwv" and "nomark" are the main items in page 5. Such a case is still acceptable, but note that in such a case the weight-averaging will not so effective to reduce the numerical noise.

In high recycling case with "passed"/"nminwv"<<1 , respawned markers cannot assign their initial weights from the weight-averaging routine. This results in the dilution of ww and wp, and sum of ww and wp changes a lot as simulation proceeds. Need to consider increasing the number of simulation markers.

- ➢ page 7 : Filter log
  - ✧ Represents the number of markers which violates the filters, and the final cause of death of the markers (by filter or boundary).
  - ✧ Also shows the number of markers which is very close to the axis (rho<rho_na), of which orbits are solved in subroutine RERK4, and the ones of which wp < 0.
  - ✧ In FORTEC-3D, when the number of dead markers accumulated to 1/10000 of the total, marker recycle routine is called. If the recycling frequency is too large ( for example, more than 1% of markers are recycled during 1 collision time), some side-effect of the filter-recycling might occur. Consider modifying the filter strength.
  - ✧ wp<0 will happens when recycling. It means that initial wp cannot be assigned in the W_AVE routine. Actually this cannot be avoided in the present version. In the future, recycling routine will be updated.

### 7.2.3 : ORBIT6

- ➢ This program converts the "orb_XXX_YYYY" files, which are generated when kido=1, to plot the guiding-center orbit trajectories.
- ➢ Usually used with parameters : kcoll=-1 (no collision), ifave=0 (no weight averaging), ifref=0 (E_r is time constant).
- ➢ Requires "xspctl.****" file created by READFLD program to convert the orbit from Boozer to Cartesian.
- ➢ On each MPI rank, ID=jist+(k-1)*jskp, k=1,2,…,jinm markers' orbit data are extracted.

Note that the initial radial position is from inner to outer surface as marker index increases.

- Execute : run "goorb6.go" using the script "orbconv.sh". Meaning of the input parameters are written in the script.
  To check the total energy conservation in case of E_r≠0 , set "lerfld=.true.".
- Plot the orbit, conservation of energy and magnetic moment : Use the example gnuplot scripts in "Orbit6/SCRIPT"
  - ✧ orb-rhoth.gpl : orbit projection on (rho, theta) plane
  - ✧ orb-enerho.gpl : rho vs (kinetic, total) energy
  - ✧ orb-chk.gpl : 3D trajectory of marker motion (color=$v_{\parallel}/v$ )
  - ✧ anim-orb.gpl : 3D animation of marker motion [color=log(|ww/wp|)]
  - ✧ orb-consv.gpl : error in (mu, kinetic energy , total energy)

### 7.2.4 : dfplot_global/xdfplot

- This program is used to plot the weight field and distribution function.
- Data output outX.{08.09,30} are generated from FORTEC-3D when ngraph≠0 and ifave=1 are given in the input parameter of FORTEC-3D.
  (Therefore, distribution function data cannot be obtained if weight-averaging is turned off.)
  - The spatial cell positions $(\rho, \theta, \zeta)$ to watch the distributions are specified by the input parameters krc,kac,kzc. To visualize $(\rho, \theta, \zeta)$-cell position in which these weight-field and distribution are watched, use "plot_cell-pos.gpl" which requires "cell_rphiz.***" file from READFLD.
- Use "sub-dfplot.sh" to run the code. Input parameter is explained in the sample.
- This code makes the following profile data
  - ✧ $\{\delta f = w \times g, f_M = p \times g, g\}$ distribution functions in two velocity spaces $(\xi, x = \frac{v}{v_{th}})$ "dffunc*.dat" , $(v_{\parallel}, v_{\perp})$ "dffun2*.dat". (* is specified in the script 'set ID="'')
    (g=marker distribution)
  - ✧ Arithmetic average and variance of weights ww and wp in $(\xi, x)$ "avewp*.dat"
  - ✧ Coefficients of average weight-field WAW(1:3) and WAP(1:3), flags for judging the average fields (idetwp) in $(\xi, x)$ "wpfld.dat".
    Note : WAW and WAP represents the fitting curve of the average fields
    $$\bar{w}(\xi, x) = WAW(1) + WAW(2) * (\xi - \xi_i) + WAW(3) * (x - x_j) \ ,$$
    where $(\xi_i, x_j)$ is the center position of (i,j)-bin in the velocity space.
    Meaning of the flag "idetwp" is written in wave2.f90.

- If ngraph>0, the data are averaged over ngraph-times in FORTEC-3D, except for WAW,

WAP, idetwp which are the immediate values. If ngraph<0, all data from FORTEC-3D are immediate value.

- When running this program, average can be taken over whole time slices contained in outX.{08.09,30}. See the meaning of "ifave" in sub-dfplot.sh.

- To make figures from the resultant data, there are several gnuplot example scripts in dfplot_global/ directory.
  Note : In the output data files, there are (0,0) dummy grid points in velocity space for convenience of gnuplot pm3d plotting.

### 7.2.5 : CHK_CONSV/gochk-consv5

- This program is used to check the conservation property of the linearized collision operator CTP and PFm_new and the local amplitude of Source/Sink term.
- Statistics data used in chk-consv is created if npltconsv>0.
- It requires GSPS library to compile. In the code, conversion of real8→real4 arrays are necessary for the plotting library. If user does not have GSPS, user can omit it.
- Input parameters in the execute script "sub-chkcom5.sh" :
  &indata1
  - ngost, ngoed : start and end ngo=X values to read the files outX.22
  - npr, npa, npz : numbers of (r, theta, zeta) cells to observe (=0 : all cells in the direction)
  - lfixnt = T/F : whether the S/S term was on/off in FORTEC-3D run
  - llocal =T/F : whether FORTEC-3D is global or local

  &indata2
  - ipr, ipa, ipz : specify the (r, theta, zeta) positions to observe

  &indata3
  - ldmode = T/F : true for observe Fourier modes of dn (density anisotropy on flux surface)
  - mmin/max, nmin/max : poloidal and toroidal modes of Fourier decomposition
  - lupl = F ; obsolete option ; not available now.

- Output data files:
  - consv.ps : postscript files to check the conservation property
  - bad-col.dat : list of cell positions of 30 largest error in the conservation of (particle number, parallel momentum, energy) in the collision term

- chkdiff.dat : error in conservation properties in CTP+Pfm, change of momentum and energy by CTP, w and p max, number of markers

  The following two are generated if ldmode=T
- dn-prof.dat : , time average of dn (density of delta-f) profile in (r,t,z) space
- dn-spctl.dat : Fourier-components of dn profile (each time slice)

  The following two are generated of lfixnt=T
- log-dmom.dat : profile of S/S term and the accuracy of S/S term (how correctly particle number, momentum, and energy were altered by S/S as intended)
- log_dmom_av.dat : Time integral of S/S term

Note : In FORTEC-3D, collision operator acting at the kc=1 cells (inner-most radial position) is different than the other part. Around the magnetic axis, markers in the poloidal cells 1<=nc<=ntheta are gathered to the nc=1 cell in the collision process to reduce statistical error in Monte-Carlo collision operater because of low population of markers near the magnetic axis. On the other hand, the CHK_CONSV plotting routine ignore the special treatment at kc=1 cells. Therefore, the reported error in the particle number, momentum, and energy (in "chkdiff.dat") looks large in the kc=1 cells, but only (kc=1,nc=1) cells are the correct ones. Usually, the conservation property of the linearized collision term is very accurate ( relative error is $< 10^{-13}$ as one can check in "bad-col.dat".)

The source/sink term is turned off at kc=1 cells in FORTEC-3D because of the low population of markers there.

### 7.2.6 : AV_FLVC-v3_4/go-findroot-v3

This program is used, after several different Er-profile runs of both ion and electron FORTEC-3D runs, to find the ambipolar solutions by smooth fittings of $\Gamma_i$ and $\Gamma_e$ on Er value. This program can be used for multiple-ion plasma cases.

How to use:

1. Before running "go-findroot", you need to finish running "go-avflvc-v3" ( § 7.2.1) and generate "Graph/ergm_fit.dat" file for each FORTEC-3D simulation run.
2. Edit "sub-find-root.sh"

   List the result directories of {species / Er-profile } cases after the line "#fortec-3d time average data paths (nspec * nfile)" as follows:

```
setenv fu11 elc-1st+-00/Graph/ergm_fit.dat
setenv fu12 elc-1st+2.5/Graph/ergm_fit.dat
```

setenv fu13 elc-1st-2.5/Graph/ergm_fit.dat

setenv fu21 ion-1st+-00/Graph/ergm_fit.dat

setenv fu22 ion-1st+2.5/Graph/ergm_fit.dat

setenv fu23 ion-1st-2.5/Graph/ergm_fit.dat

In this example, three different E_r profile simulations (+-00,+2.5,-2.5 indicate that) have been carried out for electrons and ions. Note that you can use different Er profiles for different species, and number of cases for each species can be different (two electron cases and four ion cases, for example).

Set the other environments in the script according to your environment.

3. Edit “input_findroot.dat”

1st line: number of species, radial mesh(=nercell in FORTEC-3D), Er_mesh (fineness of seeking ambipolar roots), range of seeking amb. roots [magnification factor to the Er range compared to the range of simulation runs. For example, if the Er varies [emin:emax] in the result files on a flux surface, the seeking range becomes [emin-diff : emax+diff], where diff=(Er_mag-1)*(emax-emin)/2 .]

2nd line: number of results for each species (nfile)

3rd line: number of data columns (always =4 fixed. Er, Gamma, Q, UB)

4th line: Charge of each species

5th line: order of fitting curve for each species: mfit<=nfile

Fitting curve of $\Gamma_i(\rho, E_r)$ on $E_r$ are given as (mfit-1)-th polynomials.

6th line~ : whether radially-fitted data (0) or raw data (1) in "ergm_fit.dat" is used to find root

For example : If you use radially fitted data for fitting of the electron 1st case while raw data for 2nd and 3rd electron cases (here 1st, 2nd, … corresponds to the order appeared in the list “setenv fu** elc**/Graph/ergm_fit.dat”) , and raw data for all ion cases, 6th and 7th lines becomes

0 1 1

1 1 1

4. Execute the shell script

If mfit < nfile (the order of fitting curve is smaller than the total data sets of a species), the standard variation of the fluxes, which are evaluated by go-av-flvc-v3 (written in ergm_ave.dat and qvp_ave.dat) are used as weights of errors to make fitting curves.

Output files:

amb-flux.**** :Ambipolar solutions (Er root values, and particle flux, energy flux,

parallel flows (last two have output for each species interpolated on the roots)
er-fit-flux.**** : Fitting curve data of the fluxes on each flux surfaces
msg-findroot.**** : log messages (debug purpose)

See the example gnuplot script “plot-er-flux-fit.gpl” how to see the results.

If there is no point which satisfies $\Gamma_e(\rho, E_r) = Z_i\Gamma_i(\rho, E_r)$ in the range of Er scan, this program does not tell any ambipolar solution on such a flux. On the other hand, if several roots are found, all are reported. It is recommend to use low-order fitting curves (mfit=2 or 3) even if you have many simulation cases for different Er values.

# 8. SAMPLE

In “SAMPLE/” directory, there are result files of several FORTEC-3D example runs for a LHD case, as well as the pre- and post-processes input / output are included.

NOTE: The mangetic field data created by READFLD for this case has a minor mistake. The toroidal periodicity ndivz should be 10 for LHD case, but the field data was created with ndivz=5. This does not affect the neoclassical transport calculation, but attention should be paid in plotting orbit and marker distribution in magnetic coordinates.

- ion-1st{+-00, +2.5, -2.5} : Three ion FORTEC-3D calculations with different E_r profiles ( given as “input-er. 126515t3740_H-1st.**** ). Note that the strength of Source-Sink term was increased during these simulations (ffixnt=0.4→ 0.7 from 60001 step (out2.**~).

- elc-1st{+-00, +2.5, -2.5} : Three electron FORTEC-3D calculations with the same three E_r profiles and inputs of ion parallel mean flow $\langle U_{\|i} B \rangle$ profiles (input-ub.126515t3740_H-1st.****) which are obtained by the ion simulations using AV-FLVC. Note that the ion parallel flow profiles given in the electron simulations near the magnetic axis are slightly modified from the simple time average of ion simulation results, since the ion flows have not converged in the core region. The modified $\langle U_{\|i} B \rangle$profiles are obtained by exponential fitting of the FOTEC-3D simulation result. See the last part of “dev-ergm.gpl” to learn how to obtain the exponential fitting coefficients.

- EQUIL/ : input / output of VMEC – BOOZXFORM – READFLD
- GSRAKE/ : input / output of GSRAKE for this case

In these folders, the continue files of FORTEC-3D for the last timing of ion(elc)-1st+-00 runs are contained. Note that the binary output files are in big-endian format since the simulations have been carried out on Fujitsu FX100 (based on SPARC), while the input field-data file is in little-endian, which was created on Intel PC. User must adjust the format of binary files according to your computer environment.

1. Run “SAMPLE/gather17.sh” to generate “chk-f3d.ps” and check the time evolution of the marker weights and the history of source-sink term, filter, and weight-average.

2. Run “SAMPLE/chk-markdist.gpl” to see the initial marker loading. This shows how

marker initial random distribution is relocated on each surface (total : jce=100) so that initial marker weight WP becomes as uniform as possible on each surface.

3. Run “sub-av-flvc-v34_e.sh” and “sub-av-flvc-v34_i.sh” in SAMPLE directory to evaluate time average of each simulation. In the example case, both raw 50-point radial profile and 30th-order radially-fitting profile of neoclassical fluxes are evaluated (Graph/ergm_fit.dat in each case).
   Then, use “dev-ergm.gpl” and “prof-ergm.gpl” to make figures of time evolution and radial profile of time-average of neoclassical fluxes (requires gnuplot ver. 4.6 or later). The figures are generated in “FIGURE/” directory.

4. Run “fit_result/sub-find-root.sh”. This program reads ion and electron simulation results for different E_r profiles and makes E_r-Gamma fitting curves to find ambipolar solutions. Then run “gather-ergm_fit.sh” to make “fit_result/f3d-{elc,ion}-fit-all.dat”. Finally, use “fit_result/plot-er-flux-fit.gpl” to see the result of Er-dependence fittings of Gamma, Q, <UB> and the guess of ambipolar-Er value. Note that in this case, the find-root program fails to find a root at rho=0.81~0.85 flux surfaces. You need to adjust the range of seeking the ambipolar solutions (“Er-mag” factor in “input_findroot_1st.dat”) or the order of fitting curve, or simply run new simulations with different E_r profile to add data to be used in the fitting program.

5. Sample data for CHK_CONSV and DFPLOT are prepared in “SAMPLE/ion-1st+-00_plot/”. This case continues the “ion-1st+-00” from 244001 to 245001 steps, with the parameters npltconsv=25, ngraph=50, ngo=8 (to reproduce, copy cont_007_**** and out.12,13,14 files from ion-1st+-00 and run FORTEC-3D). Then, information of conservation property of collision operator and the averaged weight field are generated every 250 steps. (Note that the output timing of weight-field data is every ngraph*ntime=50*5=250 in this example).
   Run “sub-chkcom5.sh” and “sub-dfplot.sh” (note : compiled and executed on vis1 in case of Plasma Simulator). CHK_CONSV generates Graph/consv.ps.gz figure file, as well as the Fourier spectrum of plasma density (delta-f part). DFPLOT output are all text file to the Graph directory, and one can visualize them using the gnuplot scripts in the source code directory (copied in the Graph directory). They generates distribution function of delta-f, fm, ww, wp , marker population in 4 time slices and the time averages of them.

6. Sample data for ORBIT6 is prepared in “SAMPLE/ion-1st+-00_orbit”. In the example. first 10000 steps of collision-less orbit data are sampled using 8 MPI. On each MPI rank, 30000 markers are running, but only 100 markers are sampled (total : 800 samples).

Copy and edit the script files in “PRE_POST/Orbit6/SCRIT” and run “orbconsv.sh” to analyze the orbit data. Use the ***.gpl scripts to visualize.

7. Sample data for the evaluation of neoclassical viscosity is in “SAMPLE/ion-1st+-00_ntv”. Continuing “ion-1st+-00” case from 244001 to 252001 steps, this example evaluates NTV and NPV up to 20 Fourier modes. After running ngo=8 FORTEC-3D job, run “sub-av-flvc-v34_i.sh” inside “ion-1st+-00_ntv”. Then, use “dev-ergm_ntv.gpl” and “prof-ergm_ntv.gpl” scripts in SAMPLE directory to plot the time evolution and radial profiles of NTV and NPV. Confirm that the radial neoclassical flux evaluated from the radial drift velocity moment agrees with that evaluated from neoclassical poloidal and toroidal viscosities. In “Graph/visc-mn.dat” file, the correspondence of the column in tvc_{ave,tmp}.dat and pvc_{ave,tmp}.dat with the (m,n)-components of NTV and NPV are listed.

# Appendix

## ● Appendix A : list of variables in FORTEC-3D code

( mainly those which are not appeared in the description of input parameters § 3 )

### ➢ Global variables

#### ✧ Simulation markers' variables : [ni = number of markers]

psiti, rhoi, thetai, zetai : marker position in Boozer coodinates $(\psi_t, \theta, \zeta)$. $\rho = \sqrt{\psi_t/\psi_{t-edge}}$.

roucn,uu, vz,vv,absv : marker velocity variables :

roucn= $\frac{mv_\parallel}{eB}$, uu=μ = $\frac{mv_\perp^2}{2B}$, vz=$v_\parallel$, vv=$v_\perp$, absv=$v_\parallel^2 + v_\perp^2$ (or its √)

ww,wp : Marker's weights w and p

psidt,thedt,zetdt : guiding-center drift velocity $\frac{d}{dt}(\psi_t, \theta, \zeta)$

Equilibrium field at marker's position :

bbi, eoti,cugi,cuii : $(B, \iota, G, I)$ in Boozer-coordinates

eri : radial electric field $(= -d\Phi/d\psi_t)$

dbdri, dbdti, dbdzi,dgdri, didri : gradients of equilibrium field at marker position: $(\frac{\partial B}{\partial \rho}, \frac{\partial B}{\partial \theta}, \frac{\partial B}{\partial \zeta}, \frac{dG}{d\rho}, \frac{dI}{d\rho})$

bbj, dbdrj, dbdtj, dbdzj,: up-down asymmetry part (sin-components) of magnetic field

backti, backni : background temperature and density

jr,ja,jz : $(\rho, \theta, \zeta)$-cell position of each marker (for collision, source-sink, and weight-average, jr=[1:nrcell], ja=[1:nacell], jz=[1:nzcell]

dsi, jpos, jpos1, kpos, kpos1 : coefficient and indexes used for radial interpolation of electro-magnetic field at each marker's position

ideadcnt : life time of marker

#### ✧ MHD equilibrium Spline-coefficient arrays

bco,c1bf,c2bf,c3bf : magnetic field Fourier components (cos-part)

bso,s1bf,s2bf,s3bf : magnetic field Fourier components (sin-part)

gco,c1gf,c2gf,c3gf & gso,s1gf,s2gf,s3gf : Jacobian Fourier components

cm , cn : poloidal & toroidal mode (m,n)

eot,c1et,c2et,c3et : iota spline

cui,c1i,c2i,c3i,cug ; toroidal current spline

c1g,c2g,c3g : poloidal current spline
txe,c1te,c2te,c3te : background Te spline
txi,c1ti,c2ti,c3ti : background Ti spline
dxi,c1di,c2di,c3di : background na spline (a = particle species solved in FORTEC-3D)
dln,c1ln,c2ln,c3ln : Coulomb logarithm spline
spos : radial grid positions of the Spline-fitting data
kmsh: number of radial mesh points of the Spline-fitting data

✧ Normalized constants

ratem , rateq : normalized particle mass and charge
C01~C05 : numerical factors for 4th-order Runge-Kutta
sb0[T], sx0[m], sv0[m/s], st0[s], sg0[$\mu_0$A=Tm], sEr0[V/m], sc0[C], sTem0, sep0, sflux0, sflxe0, vcnorm : normalization factors for physical quantities
almin : normalization factor for weight w and p, $\alpha_{\min} \equiv 1/\mathrm{Max}[p(t=0)]$ .
dt, dtonrm : time step sizes for collision term and guiding-center motion

Note : In the mail loop of FORTEC-3D, almost all quantites are normalized. The normalization factors are determined by sb0=B0, sx0=minor radius, sv0=v_thermal at rho=xhalf, and pmass(proton mass) or emass0(electron mass). All the other normalization factors are determined by the combination of above 4 quantities. Time step size is determined by the collision time evaluated at rho=xhalf.

✧ Phase-space meshes

nrcell, nacell, nzcell : numbers of $(\rho, \theta, \zeta)$-meshes (collision, Source/Sink, weight-average)
drcell, dtheta, dzeta, zetamax : width of the cells
nercell : number of radial mesh (to evaluate flux and viscosity)

nobx, noby : velocity-space meshes in $(\xi, \frac{v}{v_{th}})$. (weight-averaging)

vmax : max $\frac{v}{v_{th}}$ for the velocity-space

ndvx, ndvy, vmax_dist : velocity-space meshes used in DFPLOT (observation purpose)
dvol1 ,dvol2 : volume of (nrcell, nacell, nzcell) cells.
dvol3 : volume of nercell flux-annulus
rn0, rn1 : dvol*n (numbers of background plasma particles in cells)
thvr, dnty, dlm, Tr : v_th, n, lnLambda, and T at the center of nrcell-meshes.
b2ave : $\langle B^2 \rangle$ at the center of nrcell-meshes.

ic, mm : marker index and number in (nrcell, nacell, nzcell) cells.
icr, mmr : marker index and number in (nrcell) cells.
jxi,jyi,nbtt : 2D-reduced marker position in the 5D phase space cells
jxi=(v_para-pitch * theta* zeta) dimension, jyi=(velocity*radial) dimension

- Other variables contained in the modules

  - ✧ MOD_TMP
    myrank, nproc, nomp : MPI rank, total ranks, OMP threads
    ifalcX : Flags to check the allocation state of variables in subroutines
    icntX, lazst, lazed, mxst, mxed : for MPI work sharing and collective communications
  - ✧ TABLE_ERF
    Table-data of collision frequencies for the linearized collision operator
  - ✧ EVOL_ER
    epp : $=\epsilon_\perp$ , permittivity (including the classical dipole effect)
    EradX, GamiX, GameX : radial electric field and radial ion and electrion particle fluxes
    Gami1_tv,_pv : particle flux caused by toroidal and poloidal viscosity
    TVCav, PVCav, TVSav, PVSav : Fourier-decomposed toroidal and poloidal viscosity (cos-modes and sin-modes)
  - ✧ GAMMA_TABLE
    Table-data of $\Gamma(\rho, E_r)$ read from file #19
  - ✧ CHECK_BAD
    kdameX : Filter information
    diffwXXX: log of the sum of w, p, number of filtered markers, etc.
  - ✧ VAR_RECYCLE
    Variables related to marker killing & recycling
  - ✧ VSPACE
    Vecosity-space cells (diagnostic purpose)
  - ✧ VAR_MT
    Variables for random number generation

- Local variables in subroutines

  - ✧ VISC_FLUX_ER (evaluate neoclassical fluxes & viscosities)
    vir, qir, vpb : radial particle and energy flux, parallel flow UB
    dni, dnT : particle and pressure of delta-f part
  - ✧ PFm_new (Field-particle operator)

sww, spz, see : change in the velocity-moments (particle number, parallel momentum, and energy) by collision term in each cell

AA, BB: 3x3 matrix for the field-particle operator and the S/S term

WWWx,PPPx,EEEx : velocity moments in each cell

✧ W_AVE

aijX : 3x3 matrix for the averaged weight field

bbx, ccx : 3D vector of moments to be conserved by the weight averaging

WAW, WAP : fitting coefficients of the obtained averaged field

# • Appendix B: Known problems

- Simulation output (neoclassical flux, flows) tends to be noisy around the magnetic axis. The main reason is because the marker population decreases as calculation proceeds. The marker recycling routine respawn the killed markers by weight and velocity filters to the center region, but this tendency cannot be fully controlled. Initially loading more markers around the magnetic axis does not help so much.
- The relationship p*g=f_M, which is one of the key point in the two-weight delta-f scheme, is not strictly satisfied, especially near the magnetic axis. This is also due to the depopulation of markers in the core region. Because of the appearance of unusual type of orbits ( fat banana, potato, kidney), the kinetic equilibrium near the magnetic axis is no longer close to local Maxwellian. In the past, FORTEC-3D tried to fix the error in the weight P, but efficient scheme could not be established.
- Since the linearized collision operator see the background Maxwellian of which density and temperature are constant on a flux surface, the higher-order effect on the collision according to the background anisotropy is neglected. This would be important for collisional plasmas and in case ExB rotation is large (poloidal Mach number > 1), in which non-negligible dn/n0 and dnT/n0T0 are observed.
- Due to the code tuning and reducing the memory usage, the control-variate method developed by S. Matsuoka has been removed from ver. 3-3. However, the method has been proved as a powerful tool to suppress the statistical noise from the simulation outputs. Restoring the CV method should be considered in the future.

# • Appendix C: Publications related to FORTEC-3D


- "Non-local neoclassical transport simulation of geodesic acoustic mode"
  Shinsuke SATAKE, Masao OKAMOTO, Noriyoshi NAKAJIMA, et al.
  Nucl. Fusion 45 1362 (2005); https://doi.org/10.1088/0029-5515/45/11/017

- "Non-Local Simulation of the Formation of Neoclassical Ambipolar Electric Field in Non-Axisymmetric Configurations"
  Shinsuke SATAKE, Masao OKAMOTO, Noriyoshi NAKAJIMA, et al.
  Plasma and Fusion Research (2006) Volume 1, 002 ; https://doi.org/10.1585/pfr.1.002

- "Simulation studies on the GAM oscillation and damping in helical configurations"
  Nucl. Fusion 47 1258 (2007) ; https://doi.org/10.1088/0029-5515/47/9/024

- "Development of a Non-Local Neoclassical Transport Code for Helical Configurations"
  Shinsuke SATAKE, Ryutaro KANNO, Hideo SUGAMA
  Plasma and Fusion Research (2008) Volume 3, S1062 ; https://doi.org/10.1585/pfr.3.S1062

- "Benchmark test of drift-kinetic and gyrokinetic codes through neoclassical transport simulations"
  S. Satake, Y. Idomura, H. Sugama, T.-H. Watanabe
  Computer Physics Communications 181 (2010) 1069 ; https://doi.org/10.1016/j.cpc.2010.02.014

- "Neoclassical Toroidal Viscosity Calculations in Tokamaks Using a δf Monte Carlo Simulation and Their Verifications"
  S. Satake, J.-K. Park, H. Sugama, and R. Kanno
  Phys. Rev. Lett. 107, 055001 (2011); https://link.aps.org/doi/10.1103/PhysRevLett.107.055001

- "Calculation of neoclassical toroidal viscosity in tokamaks with broken toroidal symmetry"
  Shinsuke Satake, Hideo Sugama, Ryutaro Kanno and Jong-Kyu Park
  Plasma Phys. Control. Fusion 53 054018 (2011); https://doi.org/10.1088/0741-3335/53/5/054018

- "A New Simulation Method of Geodesic Acoustic Mode in Toroidal Plasmas by Using Band-Limited White Noise in a δf Neoclassical Transport Code"
  Shinsuke SATAKE, Hideo SUGAMA, Ryutaro KANNO et al.
  Progress in NUCLEAR SCIENCE and TECHNOLOGY, Vol. 2, pp.72-77 (2011);
  http://www.aesj.or.jp/publication/pnst002/data/072-077.pdf

- “Formation of Electron-Root Radial Electric Field and its Effect on Thermal Transport in LHD High Te Plasma”
  Seikichi MATSUOKA, Shinsuke SATAKE, Hiromi TAKAHASHI et al.
  Plasma and Fusion Research (2013) Volume 8, 1403039; https://doi.org/10.1585/pfr.8.1403039

- “Simulation studies of the effect of E × B rotation on neoclassical toroidal viscosity in tokamaks with small magnetic perturbations”
  S. Satake, J.-K. Park, H. Sugama and R. Kanno
  Nucl. Fusion 53 113033 (2013) ; https://doi.org/10.1088/0029-5515/53/11/113033

- “Application of an improved control-variate scheme to local neoclassical transport simulations”
  SeikichiMatsuoka, ShinsukeSatake
  Computer Physics Communications Volume 185 (2014), 2313-2321;
  https://doi.org/10.1016/j.cpc.2014.05.001

- “Experimental analyses and predictive simulations of toroidal rotation driven by the neoclassical toroidal viscosity in rippled tokamaks”
  M. Honda, S. Satake, Y. Suzuki et al.
  Nucl. Fusion 54 114005 (2014) ; https://doi.org/10.1088/0029-5515/54/11/114005

- “Effects of magnetic drift tangential to magnetic surfaces on neoclassical transport in non-axisymmetric plasmas”
  Seikichi Matsuoka, Shinsuke Satake, Ryutaro Kanno, and Hideo Sugama
  Physics of Plasmas 22, 072511 (2015); https://doi.org/10.1063/1.4923434

- “Integrated modelling of toroidal rotation with the 3D non-local drift-kinetic code and boundary models for JT-60U analyses and predictive simulations”
  M. Honda, S. Satake, Y. Suzuki et al.
  Nucl. Fusion 55 073033 (2015) ; https://doi.org/10.1088/0029-5515/55/7/073033

- “Benchmark of the local drift-kinetic models for neoclassical transport simulation in helical plasmas”
  B. Huang, S. Satake, R. Kanno, H. Sugama, and S. Matsuoka
  Physics of Plasmas 24, 022503 (2017); http://dx.doi.org/10.1063/1.4975611

- “Benchmark of the Bootstrap Current Simulation in Helical Plasmas”,
  Botsz HUANG, Shinsuke SATAKE, Ryutaro KANNO, Hideo SUGAMA, Takuya GOTO
  Plasma and Fusion Research (2017) Vol. 12, 1203004 ; https://doi.org/10.1585/pfr.12.1203004

- “Predictions of toroidal rotation and torque sources arising in non-axisymmetric perturbed magnetic fields in tokamaks”
  M. Honda, S. Satake, Y. Suzuki et al.
  Nucl. Fusion 57 116050 (2017) ; https://doi.org/10.1088/1741-4326/aa7e90

- “Effects of the applied magnetic fields with various toroidal phase differences on the neoclassical toroidal viscosity in JT-60SA”
  M. Honda, S. Satake, Y. Suzuki et al.
  Nucl. Fusion 58 112012 (2018) ; https://doi.org/10.1088/1741-4326/aabaaa