Finite-ORbit-width Transport with Electric-field and Collisions in 3D configuration
Multi-Particle-Species version

FORTEC-3D-MPS Manual

ver. X.X

(last revised, XX YY, 2022)

Shinsuke Satake and Keiji Fujita

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

1. FORTEC-3D-MPS (F3DMPS below) is a software WITHOUT ANY WARRANTY.
2. Distribution of any part of the source code is allowed only under the permission from main programmers (S. Satake and K. Fujita).
3. The Copyright of the original F3DMPS source code belongs to Shinsuke Satake (SS).
4. SS does not take any responsibility for any bugs in F3DMPS or any damage caused by the code.
5. SS never agrees with any activities to divert F3DMPS to harmful or martially purpose.
6. The users should state the name of FORTEC-3D-MPS (or FORTEC-3D) when publishing research results obtained from the code, with proper citations.
7. Any question about the code, contact SS (satake.shinsuke@nifs.ac.jp).

Contents

Chapter 1

How to compile the source code

FORTEC-3D-MPS (F3DMPS) consists of several Fortran90 source code files. In the top directly (v4-2/ or v4-2_NTC), use "make" command to compile.

Depending on the environment of the computer, one needs to adjust the compiling options in Makefile.

"KMPATH" and "LIBKM" are the paths where KMATH_RANDOM (explained below) library file exists. "depend.txt" file is created by makedepf90 command. Users do not need to update the file unless modified the source code by themselves a lot.

1.1 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 codes where you can expect speed-up by parallelizing the loops.
- FORTEC-3D requires BLAS/LAPACK libraries: Subroutine DGESV.
- F3DMPS uses "KMATH_RANDOM" library for parallel Mersenne-twister pseudo-random number generator routine. To install KMATH_RANDOM, download the source code from https://aics.riken.jp/labs/lpnctrt/projects/kmath-random/index.html and compile on your computer environment. It also requires NTL library (http://www.shoup.net/ntl/).
- Before running F3DMPS, you need to generate "jump file" for Mersenne-twister, which provides the initial parameters for the parallel PRNG routine. Read the instruction in KMATH_RANDOM manual and compile the library. Then run "km_rand_gen_jump –max_ranks 4096 –rand_range 100", for example.The generated file "jump/ file_00001" will be referred to when running F3DMPS. This jump file allows you to run up to 4096 MPI parallel run.
- F3DMPS uses MPI_COMM_SPLIT function. Check if your running environment allows you to use it.

1.2 Note for computation efficiency

1. Subroutine W_AVE in wave2_vtune.f90 assumes that there are at least 8 OpenMP threads available to parallelize (4 threads in LINEARC in linear-c-mps.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 vector-pipelining for NEC SX-AURORA (Plasma simulator in NIFS) are implemented. Since these two subroutines spend most of the computation time, it is recommended that the user tune the parameter "nppl" in module MOD_TMP (in Module-F3D_v4-2.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, Xeon Skylake : nppl~1000
Also, PHI1_EFFECT_vtune.f90, linear-c-mps_vtune.f90, and wave2_vtune.f90 are vector-tuned for SX-AURORA. Running these subroutines on other architecture, tune the parameter "nvt" and "ivlen" for better computation efficiency.

4. F3DMPS uses a lot of "mpi_in_place" to reduce the size of temporary buffer for MPI communications. However, it might be probrematic on some (old) computer environment.

5. F3DMPS uses 3 classes of MPI communicator.

   - mpi_comm_world : communicator for all MPI ranks (myrank = 0,1, . . . ,nproc-1)
   - comm_1 : communicator for each particle species
     If simulation solves 3 particle species, and nproc=120, then MPI ranks are grouped into 3 and have local communicator index (mycol1, myrank1) as follows:
     - 1st species: myrank=0,1, . . . , 39 : mycol1=0, nproc1=40, myrank1=0,1,. . . ,39,
     - 2nd species: myrank=40,41,. . . .,79 : mycol1=1, nproc1=40, myrank1=0,1,. . . ,39
     - 3rd species: myrank=80,81,...,119 : mycol1=2, nproc1=40, myrank1=0,1,. . . ,39

     MPI communication among each particle species uses the communicator comm_1. Also, outputting a data of a specific species to a file is basically done by myrank1=0 ranks. Output files are separated by particle species.
   - comm_2 : Among rank=0s of comm_1 (group of myrank1=1 ranks) In the present case, 3 ranks (myrank=0,40,80) are the member of comm_2 and have local communicator index myrank2. nproc2=3. myrank; myrank1: myrank2 = (0,0,0), (40,0,1), (80,0,2) MPI communication between different species uses the communicator comm_2.

Note: the number of ranks per a species (nproc1) is equally distributed on default, but can be controlled by setting the input parameter "nmrank()" in input-f3d***.dat. See Ch. 3.

Chapter 2

Description of input files

2.1 Input files

The number after each file name(#xx) represents of the I/O unit number to read the file. In FORTEC-3D-MPS, 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 Fujitsu FX100, written in csh script), "#PBS -v VE_FORT18=field-data.test" (on SX AURORA) or "export FORT18= field-data.test" (with Intel Fortran compiler, written in bash script).

F3DMPS is written under the policy that outout files are opened inside the program, which is different from previous single-species version. No environment variable to connect an output file name to I/O unit number is required. See Ch. 6 and 7 for detail.

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_CDF2 (§5.1) and F3DMPS runs on different computers.

In the following list of input files, the **bold character** files are mandatory ones.

- **plasma_profile_***.dat** (#10)
  - It contains density and temperature radial profile data. This file has the same format as DKES/PENTA code uses. The detail of the profile data file is explained in §5.3.
- **field-data.***** (#18)
  - It contains the data of magnetic field. Note: Different from old version of FORTEC-3D (up to ver.3.3), temperature and density profiles are not contained in the field-data file. The kinetic profiles of each species are given in a separate file (plasma_profiles file, explained above).
  - To make a field-data file, use "READFLD_CDF2" (§5.1 as well as VMEC and BOOZ_XFORM codes contained in STELLOPT package (ver.8.50 or later is recommended). Note: MPS version still can read old style of field-data file, which contains n & T profiles. The profile data are simply neglected in F3DMPS.
- gm_table.*** (#19) (required in case ifgsrk==1)
  - Table data of neoclassical fluxes Γ_a,Q_a,$\langle UB\rangle_a$, $\langle qB\rangle_a$ in (r, E_r) parameter space. It is generated from DKES/PENTA or any other neoclassical transport code.
  - The radial flux data is used in F3DMPS to solve the radial current and the time evolution of the radial electric field $\epsilon_0\epsilon_\perp \frac{\partial E_r}{\partial t} = -e\sum_a Z_a\Gamma_a$. Usually this function is used to solve ion fluxes by F3DMPS while local electron flux is given by the table data.
  - To learn how to prepare the gm_table file, see §5.2.
- Input file for F3DMPS parameters (**input-f3d***.dat**) (#91)
  - 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 Ch. 6.

- Initial radial electric field profile data (input-er.***) (#20)
  - Specifies the radial electric field profile if inief==1.
  - Details are explained in §3.2

- Ion mean parallel flow profile data (input-ub.***) (#94)
  - Give ion parallel flow $\langle U_{\parallel i} B \rangle$ profile to include the electron-ion friction force in the FORTEC-3D electron-only simulation if ifeifric==1. It is required to evaluate correct bootstrap current.
  - The $\langle U_{\parallel i} B \rangle$ profile is provided by FORTEC-3D ion simulation or any other neoclassical code.
  - Details are explained in §3.3

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

- **Jump file for KMATH_RANDOM pseudo-random number generator**
  - Path to the directory which contains the jump files (jump_00001 etc.) created by kmath_random library (explained in §1.1) should be specified by environment variable written in batch job script file to run F3DMPS. (see example gof3d-v4-2.csh).

    Example:
    setenv KMATH_RAND_JUMP_FILE_PATH /home/hoge/KMATH_RANDOM-1.1/random/ptool/jump

- Φ_1 potential profile data (out***_phi1_output.145) (#150)
  - The file is generatad by F3DMPS with Phi1 effect calculation. If ph_contin==1, then the Φ_1 profile data specified by the environment variable #150 is read when start / restart. For a normal continue job, set ph_contin=2 and then F3DMPS automatically finds the profile data file of the last run, and you do not need to specify the file name by #150.
  - Details are explained in XXX.

Chapter 3

Description of input parameters

3.1 input-f3d***.dat file

FORTEC-3D reads the "input-f3d_*.dat" assigned to file I/O unit #91 which is specified in the batch script.**

In the following list, the parameters with ☆ mark is the recommendation. Important parameters are marked with underlines.

3.1.1 namelist/read1/

Name	Type	Description
dtau	real	time step size of collision operator (dtcol = dtau*Tauc). Here, Tauc is the minimum value of collision time tau_a,b among the all combinations of particle species a and b (specified by plasma_profile file and massj,zchgj parameters in this input file), in entire radial position (r/a=0.0 to 1.0). For the accuracy of the collision term, recommended value of dtau is 1.d-4 to 1.d-5.
nss	integer	fineness of orbit calculation time step (dtorb = dtcol/nss). Note: input value of nss and actual nss inside the code is different in the MPS version. See Ch. 6.
ni	integer	marker number per MPI rank (total = ni*nproc[# of MPIs]). NOTE: ni should be divisible by the parameter nppl and nx (vector-tuning parameters). In the source codes for SX-AURORA, nppl=1000 and nx=250.
kido	integer	= 0 normal calculation. ☆ = 1 orbit check (collision and weight averaging are controlled by other parameter)

kcoll integer

= 0 no collision, orbit calculation only

= 1 ocollision (conventional like & unlike for ions + large-mass-ratio approximation for electron-ion

= 2 conventional like- + new unlike particle collisions among ions + large-mass-ratio approximation operator for electron-ion collision.☆

= 3 solve only C_ei and C_ie part (test purpose only)

= 4 Completely new operator (e-i collision is also treated by the new unlike-species operator)

ifpfm integer recommendation =2.

- (for lile-species collisions)

  = 0 neglect the field-particle collision operator (kcoll==0 case, too)

  ≥ 1 use the field-particle like-species collsion in Pfm_like with corrections for the conservation properties

- (for unlike-species ion-ion collisions)

  = 0 do not use Pfm_unlike

  = 1 use Pfm_unike w/o correction term

  ≥ 2 use Pfm_unike with correction term

- (for e-i and i-e collisions in case of kcoll=2 or 3 (use CEI_CIE)

  = 0 no corrections for weight, moment, and energy

  = 1, 2 correction according to actual change of electron momentum

  3 correction according to analytically expected change of electron momentum

npltconsv integer 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 integer 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).

nstep integer end of calculation time step.

lasym integer =.true. for up-down asymmetric case (like a single-null tokamak).

mmxcut(_as) integer 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).

ifntm,ifnpm	integer	
	= 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. NTV and NPV are not required for time evolution of E_r or Φ_1. This is only for measurement purpose.

3.1.2 namelist/cellin/

Name	Type	Description
kzmax,kamax,krmax	integer	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	integer	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. ☆

3.1.3 namelist/wavein/

Name	Type	Description
ntime	integer	the time interval to call the weight-averaging routine W_AVE every 'ntime' collision steps. Note: W_AVE requires huge amount of ALLGATHER-type communication. To reduce the communication time, set ntime =2-10 ☆ instead of ntime=1. Too large ntime would make the variance of δf distribution large and then is harmful for simulation, however.
ngraph	integer	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!) > 0 output weight fields, df, and f_M distribution averaged over ngraph times. < 0 output immediate information of the above. = 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	integer	switch of weight-averaging procedure (1: on ☆, 0: off).
ifwchg	integer	= 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).
ifrcyc	integer	switch of marker recycling 0: off, 1: do recycling☆.
irctime	integer	Do recycle procedure every irctime steps. (recommend 50 – 100). Note: irctime must be dividible by ntime.

Note: Below, filter strength parameters (***_in) have different values for each particle species (variable_in(n): n is index of particle species).

fwave_in	real	specify damping rate of weight spreading. fwave_in=F means the variance of weight will exponentially damp in 1/F collision time. Larger F means rapider damping. (recommend: fwave_in= 0.1 – 1.0), Too strong fwave results in artificial increase in effective collisionality.
idead_in	integer	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) (killed outside of the plasma by virtual limiter) , or xbnin(2)<=rhoi<=xbnout(2) (markers killed by the weight and velocity filters).
limwovp_in	real	limit of weight ww/wp (if limwovp=0 → no limit)
limwp_in	real	limit of weight wp (if limwp=0 → no limit)
limvth_in	real	limit of marker speed v/v_th (at marker position, limvth=0 → no limit)
rfadein	real	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	integer	Source-Sink term to reduce dn and dnT (0: off, 1: on☆).
ffixnt_in	real	Strength of the adaptive source-sink term of species #n (0.2-1.0) (ffixnt_in=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.
xkill	real	radial position of virtual limiter filter for markers escaping outside (usually > 1.1).
xbnin(1,2), xbnout(1,2)	real	radial positions where killed markers are respawned.

3.1.4 namelist/efldin

Name	Type	Description
ifref	integer	= 0 without radial electric field time evolution = 1 radial electric field evolution with time step size dt=dt(coll.)/nss
inief	integer	= 0 initial E_r = 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 according erfit,erfit_max,erpos,erpos_max, and eredge E_r profile if inief=-2000 can be checked using gnuplot script "SCRIPT/erprof_init.gpl". Note: if inief==-1500, E_r profile is imported from #20 even for ngo>= 1 continued job.
igsrk	integer	= 1 read gm_table file from #19. = 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	integer	= 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.

3.1.5 namelist/fun2in/

Name	Type	Description
iffun	integer	switch to select the initial marker radial distribution. = 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	integer	initial marker distribution on flux surface. = 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). (3.1) Specify proper coefficients to give radial distribution of markers (Init. marker distribution and weight wp can be seen in #41 and #42). To minimize the initial variance of the marker weight wp, the method iffun=1 is the idealistic choise because marker density is proportional to background plasma density. However, it resuls in too small population of markers near the magnetic axis. It will increase noise level of observed fluxes near the magnetic axis. Most of FORTEC-3D runs are done by simply setting iffun=2, FNA=FNB=0, FNC=1 (radially uniform initial marker distribution).

3.1.6 namelist/species

Name	Type	Description
ifelc	integer	switch of particle species to be solved. = 0 ion calculation w/o i-e collisions. = 1 electron calculation (w/o ions). = −1 all ions and electrion with i-e, e-i collisions. = −2 switch to ion-only calculation (ifelc=0) to ion+elc (ifelc=-1) (See Ch. 6).
ifeifric	integer	=1 to include ion mean flow profile in C_e,i operator. Note: In ifelc=1 case, input file input-ub.*** in #94 is required, see §3.3).
nspc	integer	number of ion species contained in the plasma_profiles_*** file (#10).
lfixne	integer =.true. adjust n_e profile according to ion density so that charge neutrality is satisfied. =.false. use ne profile from input data as it is. (Check "zchk_max" in out00x.26).	
massj(0:nspc)	integer	particle mass of jth species (rerative to proton), massj(0)=-1 (electron).
zchgj(0:nspc)	integer	charge number of jth species , zchg(0) = -1 (electron).
nmrank	integer	if user wants to vary the number of mpi ranks (nproc1) species by species, specify the ratio to nmrank(0:nspc). (see §1.2 for detail). For equally distribution, set nmrank(:)=1 ☆. Note: nmrank cannot be changed during a series of continuous jobs.

3.1.7 namelist/magf

Name	Type	Description
bmag	real	magnification factor for B-field.
cmag	real	magnification factor for collisionality.
dmag	real	magnification factor for density.
tmag	real	magnification factor for temperature.
emag	real	magnification factor of E_r [E_r(rho)=emag*input_er profile data file].

3.1.8 namelist/phi1in

Whether and how to calculate δn and Φ_1

1. **Do you want to calculate δn and/or Φ_1?**
   - YES → Go to Question 2.
   - NO → Set ph1step= 0
2. **How do you want to calculate δn and/or Φ_1?**
   - By internal calculation → Set ph1step> 0. The value of ph1step becomes the interval over which Φ_1 is averaged when updated during the internal calculation.
   - By reading from a prepared input file → Set ph1step= −1 and prepare the input file.
3. **How do you want to start the calculation of δn and/or Φ_1?**
   - From Φ_1 = 0 → Set ph_contin= 0.
   - By reading from a prepared input file → ph_contin= 1.
   - By continuing the previous job result → Set ph_contin= 2. At the beginning of a project (ngo= 0), this parameter is irrelevant, but if you will continue the job with internally calculated Φ_1, setting ph_contin= 2 is recommended for later convenience (for ngo> 0). However, if you start an internal calculation of δn and/or Φ_1 at ngo=x($\neq$ 0), you need to set ph_contin= 0 at ngo=x.
4. **Do you want to includ the Φ_1-effect in the drift-kinetic equation?**
   - YES → Set phi1_effect> 0.
   - NO → Set phi1_effect= 0.

Name	Type	Description
ph1stp	integer	Determines if and how Φ_1 is calculated. > 0 Φ_1 is internally calculated, and time average is performed every ph1stp steps. $= 0$ Φ_1 is not calculated, and all the rest in this namelist become irrelevant. $= -1$ Φ_1 is read from input file and is not updated during the calculation. The file to be read is determined by ph_contin Note: Φ_1 will not be included in the drift-kinetic equation unless the parameter phi1_effect is turned on.
ph_contin	integer	Determines if and how Φ_1 is calculated. $= 2$ Φ_1 data of the previous job is read. $= 1$ The input file specified by #150 is read.
phi1_effect	integer	Determines if and how Φ_1 is included in the drift-kinetic equation. > 0 Φ_1 is included in the drift-kinetic equation. $= 0$ Φ_1 is not included in the drift-kinetic equation. If you want to calculate δn while ignoring the Φ_1-effect in the drift-kinetic equation, set ph1stp> 0 and phi1_effect$= 0$. If ph1stp$= 0$, this parameter is forcibly set to be 0 in the code.
nlt	integer	grid number in θ direction.
nlz	integer	grid number in ζ direction.
phmode	integer	maximum mode number in the Fourier spectra of δn, and therefore of Φ_1 (see §A.1.2).
fnns	integer	fnns of 2-dimensional distribution data of δn and Φ_1.
E_consv	integer	If =1, conservation of energy and magnetic moment μ are checked. Note that the check is meaningful only if collision operator is off (kcoll=0) and E_r does not evolve (ifref=0).
Bflip	integer	If =1, the ζ-direction is inverted in the output Φ_1 profile.
wfilt	real	filter value for excluding anomalous weights in calculating δn. If $\lvert w_i - \overline{w}\rvert >$wfilt$*s_w$, w_i is exluded from the calculation, where $\overline{w}$ and s_w are the average value and standard deviation of w_i in the cell, respectively. The default value is 3.0d3.
lwwph1_log	logical	If True, a log file of anomalous weights will be output. The default value is false

3.1.9 namelist/NTCin

Name	Type	Description
wsplit	integer	Split the weight w if ≥ 1. The choice of the driving force X_1 is determined according to =1 $$X_1 = \frac{n'}{n} - \frac{ZeE_r}{T} - \frac{3}{2}\frac{T'}{T},$$ =2 $$X_1 = \frac{n'}{n} - \frac{ZeE_r}{T},$$ otherwise weight-splitting is not performed. $X_2 = T'/T$ for either choice. Note that this choice is relevant only for mapping the density variations due to each driving force and irrelevant for calculating transport coefficients (transport coefficients are converted to those for the first choice in the post process program. If you want to split the weight only for computing the transport coefficients, setting wsplit= 1 is recommended for numerical accuracy.
wrst	integer	If wrst=1, the split weights are reset (wjl=0) at the beginning of the job.

3.1.10 namelist/dblist

Name	Type	Description
nbmn_c,_s	integer	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)	real	(m,n) mode numbers of the i-th MP cos components).
smdb,sndb(i)	real	(m,n) mode numbers of the i-th MP (sin components).
ldb_c,_s (i)	real	
		ldb> 0 dbmn radial profile (prop. to rho**ldb) [ldb is real number]
		ldb≤ 0 bmn(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              2,10  !original 3rd mode
      6     3,10              2, 0  !original 4th 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 $\langle U_\parallel B \rangle(\rho)$ [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
.....       .....
0.995     -111.33
(total nercell radial points, on half-mesh)
```

Note: if there are several ion species, repeat the radial profile data as above style for all ion species (w/o blank lines between species)

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)
```

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 the output in out*.26.

Chapter 4

Description of each subroutine

4.1 Initialization

- KMATH_Random_Init, load_grnd_dsfmt, KMATH_Random_Seed
  —Setup Mersenne-Twister random number generator
- READ_BFIELD_TABLE
- READ_VOL_TABLE
- READ_GAMMA_TABLE
- READ_NTPROF
  —Read field-data and gamma-table, and density, temperature profile data
- SPLIT_COMM, COMM_RANK0S : define communicators for MPS simulation
- NTRTBL : make radial table of n, T, and tau_jk etc., and define time step size
- CLOG_MPS : Coulomn log value for MPS simulation
- COL_TABLE : Order of the pair of collision calculation in MPS simulation
- SET_PARA_RANGE : Set parameters for MPI work-share and communications
- ALLOC_VAR : allocate arrays decrared in modules
- DFPPRE : preparation for PLOTWGT and DFPLOT
- TBLERR_MPS : prepare error-function and some other table data used in the collision operator
- RNxCALC : evaluate cell volume*density in (r,theta,zeta)-cells.
- ERAD_PRE : preparation for interpolation of E_psi profile
- pre_gauss2 : preparation for table used to generate Gaussian distribution
- INIT_ERPROF : setup initial E_r profile
- MARKDIST , INIT_WP : initial marker loading and initial weight
- RW_JOB_DATA : read and write continue file

4.2 Main loop

- ORBIT
  - VISC_FLUX_ER : Evaluate transport and viscosity
    - SOLVE_DERDT : solve 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 to reduce noise
- RECYCLE_MARKER : marker recycling
- LINEARC_MPS : collision term
  - CALC_UBI : evaluate <UB> of ion species (ifelc=-1 case)
  - CEI_CIE : e-i collision (pitch-angle scattering + friction b/w elc-ion)
  - CTP_LIKE, PFm_LIKE : Test- and field-particle operators (like-species)
  - CTP_UNLIKE, PFm_UNLIKE : Test- and field-particle operator (unlike-species)
  - SOURCE_SINK : Source-Sink term
- W_AVE : Weight averaging to smooth the weight field
  - PLOTWGT, DFPLOT : export data of weight field and distribution function
- BAD_LOG : export filter and recycling information
- WRITE_VISCS, WRITE_ERGM : output data of viscosity and fluxes

4.3 Φ_1 and δn

- PHI1_DATA : determine if and how Φ_1 is calculated and setup the Φ_1 calculation.
  - PHI_READ : read input Φ_1 file.
  - PHI_ALLOC : allocate variables for the Φ_1 calculation.
- CHK_PROF : calculate coarse-grained spatial distribution functions (density profiles) on each flux surface and perform Fourier transformation to obtain the density spectra.
- EVAL_PROF : construct Φ_1 from the density variations evaluated in CHK_PROF, call SPL_Phi1 to perform spline interpolation in the radial direction, and output data for drawing Φ_1 and δn profiles.
  - SPL_Phi1 : perform spline interpolation of Φ_1 to make it continuous in the radial direction.

4.4 Transport coefficients

- SET_NTC : Initialize the transport coefficients calculation.
- SET_FORCE : Calculate diving forces.
- Wij_CHANGE : Convert the split weight w_{ab}^j to/from self-components w_{aa}^j, where a and b are species indexes and j is the force index.

4.5 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
- WRITE_TIME_LOG : estimate time to finish the present job
- grnd_dsfmt : Mersenne-Twister random number generator
- save_grnd_dsfmt : save Mersenne-Twister parameters to continue job
- 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
- FUN1, FUN2 : functions used to give initial radial distribution of markers
- BAD_LOG_SUM : create summary of out*.15,16, and 17 (all species)

Chapter 5

How to set up the input files

5.1 Creating field-data file

Requires VMEC, BOOZ_XFORM, and READFLD2_CDF 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
- READFLD2_CDF(written by Satake) : Translator from boozmn file to the field-data file for FORTEC-3D.
  Note :
  - Presently READFLD2_CDF 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).
  - READFLD2_CDF provides the output files which contains the information plotted by the library so that users can check the data by your own way. Different from old version of READFLD (for single-species version FORTEC-3D), READFLD2_CDF does not prepare n & T profile data nor input data for GSRAKE.
  - READFLD2_CDF calculates the "U_tilde", which is used to evaluate Pfirsch-Schluter flow, but it does not contained in field_data file presently. It will be utilized in futute.

Because of easiness of handling the data, netCDF format of wout and boozmn files are preffered to use READFLD2 and FORTEC-3D programs. Text-binary style data is planned to be excluded in future version.

For more details, read "vmec-boozxform_eng.pdf" and "readfld2_geom_data.pdf"

5.2 Creating gm_table file

"gm_table" data file has the table data of radial neoclassical particle and energy fluxes, $\Gamma_{i,e}$ and $Q_{i,e}$, $\langle U_\parallel B\rangle_{i,e}$, $\langle q_\parallel B\rangle_{i,e}$ in the parameter space (ρ, E_r), ($Q_{i,e}$ and $\langle q_\parallel B\rangle_{i,e}$ are optional). The Γ_e-table is required to run a E_r time-evolution simulation of F3DMPS when only ion is solved by F3DMPS (ifelc=0 and ifref=1 case).

Prepare $\Gamma_{i,e}$ (and $Q_{i,e}$, $\langle U_\parallel B\rangle_{i,e}$, $\langle q_\parallel B\rangle_{i,e}$ if possible) table data by any other codes such as DKES, DKES/PENTA, DGN/LHD, GSRAKE, or KNOSOS, 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 $\Gamma(\rho, E_r)$-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 $\Gamma(\rho, E_r)$-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 Density and temperature radial profile data (plasma_profile file)

Input data of radial density and temperature profiles for F3DMPS is given by a table of n & T values on flux surfaces, "plasma_profiles_***.dat" file, which is connected as unit #10 of I/O when F3DMPS is running.

The format of the file is the same as that used for PENTA code. Examples are shown below :

```
-----------------
101
0.000  4.00e+01  5.00e+03  2.00e+01  3.00e+03  1.000e+01  3.00e+03
0.010  3.90e+01  4.90e+03  1.95e+01  2.95e+03  0.975e+01  2.95e+03
0.020  3.80e+01  4.80e+03  1.90e+01  2.90e+03  0.950e+01  2.90e+03
...
0.990  0.60e+01  1.10e+03  0.30e+01  1.55e+03  0.150e+01  1.55e+03
1.000  0.50e+01  1.00e+03  0.25e+01  1.50e+03  0.125e+01  1.50e+03
----------------------
```

The first line is the number of radial data points contained in the file. From the second line, the columns represents radial position (r/a), (n, T) (species 0) (n, T) (species 1), ... The 0's species should always be electron, even if you do not solve electron in F3DMPS. This example is for electron, hydrogen, and helium plasma. The number of species contained in the profile data file should be equal to nspc+1, where nspc is given in "input-f3d***.dat" file.

Density and Temperature units should be [1/m^3] and [keV], respectively. In the example above, density is given in 1.e+18/m^3 unit and temperature in eV unit. Then, you can adjust the unit by using "dmag" and "tmag" parameters in "input-f3d***.dat" file. For example, "dmag=1.0d18, tmag=1.0d-3" in this example case.

Information of atmic mass and charge of each species are specified by "massj(0:nspc)" and "zchgj(0:nspc)" in "input-f3d***.dat".

Note:

1. The profile data should contain r/a=0.00 and r/a=1.00 radial position data.
2. The radial mesh points can be either equal interval or arbitral interval. The radial profile then interpolated in F3DMPS by 3rd-order spline function.

3. Charge neutrality (Sum_a [Z_a*n_a] =0) is not required to strictly satisfied in the profile data. You can choose either

   - use the input n_a profile as it is, neglect the charge neutrality, or
   - correct electron density according to given ion profile so that exact charge neutrality is satisfied. This is done if "lfixne=.true." is given in "input-f3d***.dat" n_e profile before and after the adjustment can be checked by the output of "out000.26".

4. The density and temperature data read and used in F3DMPS simulation are written in "out_YYY.12", where YYY is the species index (0 to nspc).

5. Temperature of each species can be different, not only T_e and T_i, but also among ion species. If "kcoll=2", Sugama's new collision operator for different ion temperature is used, and if "kcoll=4", all electron and ion collisions are treated by the new operator. However, kcoll=4 has not been tested well and may cause error.

Chapter 6

How to run FORTEC-3D-MPS

6.1 Examples

There are two examples of simulations that ran on JFRS1 of IFERC-CSC.

1. 3 ion species (global) + electron (using local NC simulation table data), solve time evolution of ambipolar-Er. (EXAMPLE/3ion-erevolv)
2. 3 ion species + electron, all were solved by F3DMPS, at a fixed E_r profile. (EXAMPLE/3ion+elc) Both cases use a LHD magnetic field and kinetic profile same as those used in [K. Fujita, JPP2020].

Note: These examples are too small (number of simulation markers and time steps) to obtain a physically reasonable result. In a product run, the number of markers per species (= ni * [# of MPI rank per species] should be $> 5 \times 10^7$ and continue the job up to more than 3 collision times.

EXAMPLE/3ion-erevolv

- It runs 2 continuous jobs using 20 threads × 60 MPI (20 × 3 species) = 1200 cores, 2000 steps each. This example also contains NTV and NPV calculation.

1. Edit "gof3d-v4-2.csh" batch job script file according to your computer environment. Submit the first job. When the job finishes, check the last line of "out000.26" is as follows:

```
END of calculation : nstep =    2000
```

2. Then, edit "input-f3d-v4-2.dat" as follows:

```
nstep=4000, ngo=1
```

   and submit the job again.

3. When the second job ends, the last line of "out001.26" should be

```
END of calculation : nstep =    4000
```

4. For post-process, use "av-flvc", "chk17_mps.gpl", "chk46.sh", which are explained in §7.2~.

EXAMPLE/3ion+elc

- It runs 2 continuous jobs using 20 threads × 100 MPI (40 for H+, 20 for electron, He2+, C6+) = 2000 cores, 1000 steps each. This is an example of unequal distributed MPI ranks per species in order to reduce the noise level of H+ calculation.

How to run : similar to the case above.

6.2 Setup rule of the time step size in FORTEC-3D-MPS

The rule to set up the time step size in MPS version has changed from previous, single-species version. Because all species have to use the same time step size for collisions and E_r evolution, the time step size is determined in the code as follows in subroutine NTRTBL:

1. The control parameters of time step size are "nss" and "dtau" in input-f3d***.dat file.
2. In the beginning of the code, the minimum collision time, Taumin = $\mathrm{Min}[\tau_{\mathrm{a,b}}(\rho)]$, is evaluated for all the combinations of $\{a, b\}$-species collisions (including like-species collisions).
3. Then, the time step size for Coulomb collision operator, dtcol, is defined as
$$\mathrm{dtcol} = \mathrm{dtau} * \mathrm{Taumin} \tag{6.1}$$
4. Next, for each species i, evaluate the following quantity on every flux surface.
$$\mathrm{dlc}(\rho)_{\mathrm{i}} = \mathrm{R}_0/(\iota(\rho) \times \mathrm{v}_{\mathrm{th,i}}(\rho) \times \mathrm{dtcol} \times \mathrm{ndivz}) \tag{6.2}$$
This quantity measures about how many time steps a passing particle of species i will spend to pass one toroidal period along a field line, if the orbit time step size is the same as dtau.
5. Find the miminum value of $\mathrm{dlc}(\rho)_{\mathrm{i}}$ (among all i and ρ). Then, using the minimum value mindlc, the time step size for orbit time integration, dtorb, is defined as follows:

```
nss_in = nss   ! ← input value of nss
nss_tmp = int(dble(nss_in)/mindlc) +1
if(nss_tmp<4) then
   nss=nss
else
   nss=nss_tmp
end if
dtorb = dtcol/dble(nss)
```

This means, the time step size dtorb is determined so that even the fastest (thermal velocity) passing particle spends at least nss(input)+1 time steps to pass one toroidal period along a field line. Information of the defined time step sizes appears in out_YYY.12.

Note:

- From the rule above, one can find that dtorb becomes very small when electron is solved.
- Recommended choise : dtau =1d-4 ~ 1d-5, nss> 20 (for LHD), > 50 (for tokamak). Note that dtorb depends on the toroidal period number of the magnetic configuration, ndivz.
- If one switches ion-only calculation (ifelc=0) to ion+electron calculation (ifelc=-2), the time step sizes dtcol and dtorb are redifined when continueing a job.

Chapter 7

How to see the results (post-process)

7.1 Description of output files

Note: Here, X indicates the ID of continuous jobs and Y indicates the particle species index. "bin" means binary file

- outXXX_YYY.106: General information of the job for species YYY. User can see the progress of job during it is running by watching this.
- outXXX_YYY.07 : Radial profiles of E_r and Gamma_d,_vnc (radial flux evaluated from radial drift velocity v_dr and viscosity, respectively) at each time steps, and Gam_tv, Gam_pv , which are fluxes proportional to NTV and NPV (see Satake, PPCF 2012). If a calculation has enough accuracy, Gamma_d and Gamma_vnc should coincide with. This file is created only if ifntv≠ 0 or ifnpv≠ 0.
- outXXX_YYY.08, 09, 30 (bin): Profiles of delta-f distribution function, average weights, and interpolated weight fields, respectively. This file is created only if ngraph≠ 0. To read these files requires a post-process code DFPLOT (§*.*).
- out_YYY.12: Radial profiles of density, temperature, collision time b/w species YYY and others, time step size, typical orbit step size, etc.
- out.13: Radial profiles of density, temperature, Coulomb log., error-function table (used for collision operator so on), initial E_r profile. out_YYY.12 and out.13 are created only when ngo=0. (note: out_YYY.12 are created again when ifelc=-2, and old ones are renamed as out_YYY_ion.12)
- out_YYY.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_YYY.14.
- outXXX_YYY.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.
- outXXX_YYY.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.
- outXXX_YYY.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.
- out.15, out.16, out.17 : summaries of outXXX_YYY.15,16,17 These files are updated when job is continued. To plot the logs for filters, use "chk17_mps.gpl" (§7.2.2)
- outXXX_YYY.22 (bin) : Log file for Linearized collision operator and adaptive source-sink term. This file is created only if npltconsv>0. To read this file requires "CHK_CONSV_MPS" code (§*.*).

- outXXX_YYY.27 : Radial profiles of Gamma, energy flux Q, parallel flow, flux-surface averaged density and pressure perturbation dn/(dn0), dnT/(dnT0) at each time steps. The smallest YYY number file (000 if ifelc/=0, or 001 otherwize), it also contains Sum_i (Z_i*Gamma_i) -Gamma_e. Note : This Gamma_e is either the one F3DMPS solves (ifelc/=0) or from gm_table (ifelc=0 and ifgsrk=1).

- outXXX_YYY.67, 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: outXXX_YYY.63 and .64 are also produced.

- outXXX_YYY.107, 108 (bin): New format of Er, Gamma, Q, <UB>, <qB>, dn, dnT evolution data. These files are intended to reduce the time to read outXXX_YYY.07 and 27 text-style files in order to speed up the post-process code AV-FLVC. (§7.2.1)

7.1.1 Transport coefficients

- outXXX_YYY.127 (bin): Output file of transport coefficients D_{ab}^{kl} and driving forces X_b^l.

7.1.2 Φ_1 and δn

- outXXX_YYY.140: Filter log of $\Phi_1/\delta n$ calculation. Opened only if wwph1_log=.true..

- outXXX.141: Φ_1 profile read at the beginning of a job (basically for debugging purpose).

- outXXX.142: Mapping data of Φ_1 and δn:

```
# r-mesh = 2
0    0  -0.4163430E-03  -0.1078863E+01   0.1569121E+17  -0.2004314E+16  ...
1    0  -0.1302041E-03  -0.3373956E+00   0.2451815E+17   0.6124268E+15  ...
2    0   0.1210629E-03   0.3137084E+00   0.3135337E+17   0.2800754E+16
3    0   0.3083219E-03   0.7989494E+00   0.3571781E+17   0.4307737E+16
4    0   0.4105807E-03   0.1063931E+01   0.3753603E+17   0.4960824E+16
5    0   0.4166340E-03   0.1079617E+01   0.3709615E+17   0.4684863E+16
6    0   0.3254854E-03   0.8434250E+00   0.3495535E+17   0.3507747E+16
7    0   0.1455703E-03   0.3772138E+00   0.3180744E+17   0.1554416E+16
8    0  -0.1070477E-03  -0.2773910E+00   0.2833453E+17  -0.9696665E+15
9    0  -0.4112064E-03  -0.1065552E+01   0.2506690E+17  -0.3803228E+16
10   0  -0.7435245E-03  -0.1926683E+01   0.2227410E+17  -0.6658932E+16
.    .
.    .
.    .
```

  From the left, each column corresponds to θ-mesh, ζ-mesh, $e\Phi_1/T_i$, Φ_1 [V], $\delta n_I, (I = 1, 2...)$ [$1/m^3$], where I refers to ion species.

- outXXX.143: Mapping data of split δn. This file will be output only if wsplit> 0.

- outXXX.144: Profile of $\Phi_0(r) = -\int^r E_r(r')dr'$ at the beginning of the job.

- outXXX_phi1_output.145: Φ_1 and δn data for continuing the internal $\Phi_1/\delta n$ calculation.

- outXXX_YYY.146: Monitoring data of $\Phi_1/\delta n$ calculation.

- outXXX.147: Monitoring data of $\Phi_1/\delta n$ calculation. Common for all species.

- outXXX.148: Spectrum data of $\Phi_1/\delta n$ for plotting.

- outXXX_YYY.152: Log file of energy conservation. Output only if E_consv=1,

7.1.3 Others

- out_YYY.41, 42, 43 : Initial marker distribution and initial marker weight information. To plot, use "chk-markdist.gpl". (§7.2.8)

- outXXX_YYY.46 : Check log for the weight-averaging. To plot, use "ckh46.sh" (§7.2.3)

- outXXX.98 : Record of the parameters used in the XXX job.

7.2 Post-process programs

7.2.1 AV-FLVC/go-avflvc-v4 (for F3DMPS ver.4-2 , use AV-FLVC ver.4-2 or later)

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

Note:

1. This program can run thread parallel, compiler's auto parallel and openMP. The openMP parallel part is only when the output files are read. (for quick reading many data files). User can choose the compiler's option to activate openMP or not, depending on whether thread-parallel file reading is effective or not on user's envilonment.

2. 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.

Examples of averaging setup parameters are given in "input-av-flvc-v4-2.txt". in the EXAMPLE directories. You can evaluate several ways of average.

&param

Name	Type	Description
ntmp	integer	step size of short-term averaging to see the evolution (outputs to ***_tmp.dat")
nstart , nend	integer	timing for long-term average (outputs to ***_ave.dat)
ngost, ngoed	integer	start and end job ID to be read in the post-process. • Ex: ngost =2, ngoed=4 → read out002,003,004_YYY.107 etc. • Note: the nstart, nend range should be contained in the jobs specified by ngost and ngoed.
ngoswc	integer	if you have switched ifelc=0 to ifelc=-2 (ion-only to ion+elc simulation), give the ngo job number when electron calculation has begun. set ngoswc=0.
ntv, mpv	integer	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 (m,n) modes contained in #67, #68 etc. Note: 1. If there are more than 50 modes, output is truncated and the sum of the truncated modes are written. 2. If the number of NTV (NPV) modes (ifntm, ifnpm in input-f3d***.dat) are larger than ntv, mpv, output in tvc_tmp.dat etc. are truncated up to ifntm modes, etc. In such a case, the truncated NTV (NPV) modes (ntv+1, ntv+2, . . . , ifntm) are summed up and appears in the last (right most) column in tvc_tmp.dat etc.
ifnorm	integer	=1 to normalize NTV and NPV by local pressure gradient (see the source code av-flvc-v4-2.f90 for details)

&norm

Name	Type	Description
lsmooth	logical	.true. if you need smooth-fitting curves for Gamma, Q, UB, NTV, and NPV (outputs to ***_fit.dat)
mfit	integer	order of the polynomials for the fitting curves (3<=mfit<nercell)
nrfit	integer	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.

&stpin

Name	Type	Description
nstep	integer	`= 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 from ngo=0 to ngoed.

To plot the data, please use "dev-ergm_mps.gpl", "prof-ergm_mps.gpl" Gnuplot script files for example.

7.2.2 chk17_mps.gpl

This script makes log plots of quick "health check" for FORTEC-3D simulation run whether the simulation runs correctly, by watching the change in the markers' weights. Timing of writing data to outXXX_YYY.17 is controlled by the input parameter "ifwchg". out.17 is summary for each job (ngo=0,1,. . .).

This gnuplot script also plots out.15 (killed marker information) and out.16 (filter log).

Note: From v4-1, out***.15,16,17 are two types :

- Detailed log for each job, each species : outXXX_YYY.15,16,17
- Summary of all job, all species : out.15,16,17

chk17_mps.gpl plots only the latter ones.
To see the details of the logs, see outXXX_YYY.15,16,17. Plotting these files with "chk15_16_17_mps.gpl".

How to use:

1. Assume two simulation data files are /work/case1, /work/case2.
2. Put chk17_mps.gpl to /work/.
3. Edit chk17_mps.sh. Write the directory name list into "do for [CASE in " . . . "] {
   Ex. : do for [CASE in "case1 case2"] {
4. Execute "gnuplot chk17_mps.sh. It generates chk17_***.ps" under /work/caseX/ directories
5. The meaning of each figure is as follows:
   - page 1 : ratio of particle numbers (per species, steps, total particle numbers per species) killed by edge limiter and filters. These quantities should be very small, < 1.e-7.
   - page 2, 3 : rate of particle numbers which hits several filters:
     - page 2 : watch strange behavior
       * psi<0 : strange orbit near the axis (contains case rho¡rho_na, not error)
       * absv<0 : absolute v becomes <0 after collision
       * vpp<0 : v_perp<0 after collision
       * ptc> : —v_para/v—>1 after collision
       * wp<0 : weight p<0 after weight average
     - page 3: filters
       * ww/wp> : w/p>limwovp
       * wp> : p>limwp
       * wvz> : —w*v_para—>limwvz
       * v/vth : v/v_th > limvth
   - page 4 : sum of weight ww and wp. Check ww/wp $\ll$ 1.
   - page 5: Change in weight w and p. "RK (Runge-Kutta), flt (filter), RCL (recycle), S/S (source/sink)" represents the breakdown of the cause of change in weight.
     Note:
     - Positive value is plotted by lines, negative by points
     - Since d(w+p)/dt should be zero in RK, two RK lines and points shoud overlaps each other.
     - Change in weight w and p by filters and S/S should be very small compared to the total sum of w and p.
   - page 6 : Change in weight w and p by wave (weight average) and col (collision). These quantities should be round-error level. (<1e-10)

7.2.3 chk46.sh

This script plots outXXX_YYY.46, which is the log for weight-average routine, and generate "Graph/chk46_all.ps" Description of the plots:

- Total numbers of cells in phase space (nrcell*nzcell*nacell*nobx*noby)
- 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 are no existing ones to evaluate average weight field.
- dgesv: error in matrix solver DGESV (singularity happens or some other reason)
- 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$\gg 0$: 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 the page "log(2)" is very few. It is a preferable situation that the "passed" item is close to (at least more than 1/10 of) the "Total". If marker population is very low compared to the total number of the bins, "nminwv" and "nomark" are the main items in page "log(1)". 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"$\ll 1$, 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.

7.2.4 ORBIT6 (under construction)

- 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.

7.2.5 dfplot_global/xdfplot (under construction)

- 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$\neq 0$ 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.)

7.2.6 CHK_CONSV_MPS/gochk-consv_mps (under construction)

- 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.

7.2.7 AV_FLVC/go-findroot-mps (under construction)

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 Γ_i and Γ_e on Er value. This program can be used for multiple-ion plasma cases.

7.2.8 chk-markdist.gpl

This script plots the data in out_YYY.41,42,43, to check the initial simulation marker loading.

- page 1: F3DMPS loads markers not simply using random numbers, but adjust the marker initial positions so that the variance in marker population per unit volume on each flux surface is small enough. First page is the log of the variance and number of cells which are not converged.
- page 2: radial profile of average wp and its variance on the flux surface.
- page 3- (total jce=100 pages): marker population on a flux surface, each (θ, ζ)-cell (total: ntheta*nzeta). Initial = initial loading by random numbers, last: after adjustment of loading position, expect: expected variance of initial loading population.

Appendix A

Simulation model

A.1 Calculation of Φ_1

A.1.1 How to evaluate Φ_1

Assuming the adiabatic response to Φ_1, the lowest order density becomes

$$n_{a0} \to n_{a0} \exp\left(-Z_a e\Phi_1/T_a\right). \tag{A.1}$$

Thus, the total density up to the first order becomes

$$n_a = n_{a0} \exp\left(-Z_a e\Phi_1/T_a\right) + \delta n_a. \tag{A.2}$$

When the assumption

$$Z_a e\Phi_1/T_a \ll 1, \tag{A.3}$$

holds, we can expand (A.2) as $n_a \simeq n_{a0} - (Z_e\Phi_1/T_a)n_{a0} + \delta n_a$. Then, substituting this expression into the quasi-neutral condition, $\sum_a Z_a n_a = 0$, gives

$$\begin{aligned} -e^2\Phi_1 \left(\frac{n_{e0}}{T_e} + \sum_I Z_I^2 \frac{n_{I0}}{T_I}\right) \\ + e\left(-\delta n_e + \sum_I Z_I \delta n_I\right) = 0. \end{aligned} \tag{A.4}$$

where the subscript I refers to ion species. From this, we obtain an expression

$$\begin{aligned} \Phi_1 &= \frac{1}{e}\left(\sum_I \frac{Z_I^2 n_{I0}}{T_i} + \frac{n_{e0}}{T_e}\right)^{-1} \sum_a Z_a \delta n_a \\ &= \frac{1}{e}\left(\sum_I \frac{Z_I^2 n_{I0}}{T_i} + \frac{n_{e0}}{T_e}\right)^{-1} \sum_a Z_a \int d^3v \delta f_a \end{aligned} \tag{A.5}$$

Therefore, the structure of Φ_1 is determined by the density variation δn_a.

Note: Since the assumption (A.3) does not hold for high-Z impurities, FORTEC-3D needs to be updated to calculate Φ_1 by a more general equation.

A.1.2 Fourier spectrum

The Fourier coefficients of a function $A(\boldsymbol{X}, \boldsymbol{v})$ are defined as

$$\tilde{A}^{0,0}(r, \boldsymbol{v}) = \frac{1}{4\pi^2} \int\int d\theta d\zeta A(\boldsymbol{X}, \boldsymbol{v}), \tag{A.6}$$

for $m = n = 0$, and otherwise as

$$\tilde{A}^{m,n(c)}(r, \boldsymbol{v}) = \frac{1}{2\pi^2} \int\int d\theta d\zeta A(\boldsymbol{X}, \boldsymbol{v}) \cos(m\theta - Nn\zeta), \tag{A.7}$$

$$\tilde{A}^{m,n(s)}(r, \boldsymbol{v}) = \frac{1}{2\pi^2} \int\int d\theta d\zeta A(\boldsymbol{X}, \boldsymbol{v}) \sin(m\theta - Nn\zeta), \tag{A.8}$$

where m and n denote the poloidal and the toroidal mode numbers, respectively, and N is the toroidal period number ($N = 10$ for LHD).

In FORTEC-3D, the mode ranges are determined by the input parameter phmode (see §3.1.8) as

$$-\mathsf{phmode} \le m \le \mathsf{phmode}, \quad 0 \le n \le \mathsf{phmode} \tag{A.9}$$

Note: In FORTEC-3D, $\tilde{A}^{m,0(c)}$ and $\tilde{A}^{m,0(s)}$ modes ($m \neq 0$) are duplicately produced by the Fourier transformation process because the corresponding DO LOOPS run for the entire mode ranges (A.9), while $\tilde{A}^{|m|,0(c)} = \tilde{A}^{-|m|,0(c)}$ and $\tilde{A}^{|m|,0(s)} = -\tilde{A}^{-|m|,0(s)}$.

In the code, for computational efficiency, the duplication is coped with by normalizing the modes as

$$\tilde{A}^{m,0(c)} \rightarrow \frac{1}{2}\tilde{A}^{m,0(c)}, \tag{A.10}$$

$$\tilde{A}^{m,0(s)} \rightarrow \frac{1}{2}\tilde{A}^{m,0(s)}, \tag{A.11}$$

instead of setting the redundant modes to zero. As a consequence of this treatment, the output values of the modes are halved. For example, let us suppose the actual value of $\tilde{\Phi}_1^{1,0(s)}(r)$ is 2.0×10^{-2}. Then, the output values becomes

$$\tilde{\Phi}_1^{1,0(s)}(r) = 1.0 \times 10^{-2}, \tag{A.12}$$

$$\tilde{\Phi}_1^{-1,0(s)}(r) = -1.0 \times 10^{-2}. \tag{A.13}$$

Therefore, when the user checks or plots the values, do not forget to double the relevant modes.

Appendix B

Lists

B.1 list of variables in FORTEC-3D-MPS

(mainly those which are not appeared in the description of input parameters, Ch. 3).

B.1.1 Global variables

Particle species information / MPI in MPS version

Name	Type	Description
my_j	integer	index of species a MPI rank solves.
nspc	integer	number of ion species.
nspl	integer	total number of species actually solved in a simulation.
kst	integer	smallest index of species (my_j=kst,kst+1,. . . ,kst+nspl-1).
massj, zchgj	integer	mass and charge number of particle species.
amasj, achgj	real	real*8 value of mass and charge.
myrank	integer	global MPI rank in MPI_COMM_WORLD (0 ≤myrank≤nproc-1)
myrank1	integer	local MPI rank in a communicator for each species, comm_1 (0 ≤myrank1≤nproc1-1).
myrank2	integer	MPI ranks in a communicator among myrank1=0 ranks, comm_2 (0 ≤myrank2≤nproc2-1).

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

Name	Type	Description
psiti, rhoi, thetai, zetai	real	marker position in Boozer coordinates (ψ_t, θ, ζ). $\rho = \sqrt{\psi_t/\psi_{t-edge}}$.
roucn,uu, vz,vv,absv	real	marker velocity variables: roucn= $(mv_\parallel)/eB$, uu = μ = $(mv_\perp^2)/2B$, vz= $v_\parallel$, vv= $v_\perp$, absv= $v_\parallel^2 + v_\perp^2$ (or = $\sqrt{v_\parallel^2 + v_\perp^2}$).
ww,wp	real	Marker's weights w and p.
psidt,thedt,zetdt	real	guiding-center drift velocity $d(\psi_t, \theta, \zeta)/dt$.

Equilibrium field at marker's position:

Name	Type	Description
bbi, eoti,cugi,cuii	real	(B, ι, G, I) in Boozer-coordinates
eri	real	radial electric field $(= -d\Phi)/(d\psi_t)$.
dbdri, dbdti, dbdzi,dgdri, didri	real	gradients of equilibrium field at marker position: $(\partial B/\partial\rho, \partial B/\partial\theta, \partial B/\partial\zeta, \partial G/\partial\rho, \partial G/\partial\rho)$.
bbj, dbdrj, dbdtj, dbdzj	real	up-down asymmetry part (sin-components) of magnetic field
backti, backni	real	background temperature and density

Name	Type	Description
jr,ja,jz	real	(ρ, θ, ζ)-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	real	coefficient and indexes used for radial interpolation of electro-magnetic field at each marker's position.
ideadcnt	integer	life time of marker

MHD equilibrium Spline-coefficient arrays

Name	Type	Description
bco,c1bf,c2bf,c3bf	real	magnetic field Fourier components (cos-part)
bso,s1bf,s2bf,s3bf	real	magnetic field Fourier components (sin-part)
gco,c1gf,c2gf,c3gf & gso,s1gf,s2gf,s3gf	real	Jacobian Fourier components
cm , cn		poloidal & toroidal mode (m,n)
eot,c1et,c2et,c3et	real	iota spline
cui,c1i,c2i,c3i,cug	real	toroidal current spline
c1g,c2g,c3g	real	poloidal current spline
txi,c1ti,c2ti,c3ti	real	background T spline of a species my_j
dxi,c1di,c2di,c3di	real	background n spline of a species my_j
spos	real	radial grid positions of the Spline-fitting data
kmsh	integer	number of radial mesh points of the Spline-fitting data

Normalized constants

Name	Type	Description
ratem , rateq	real	normalized particle mass and charge
C01-C05	real	numerical factors for 4th-order Runge-Kutta
sb0[T],sx0[m],sv0[m/s],st0[s],sg0[μ_0A=Tm], sEr0[V/m],sc0[C],sp0[V],sm0[kg],sTem0, sep0,sflux0,sflxe0,vcnorm	real	normalization factors for physical quantities
almin	real	normalization factor for weight w and p, $\alpha_{\min} \equiv 1/(\mathrm{Max}[p(t=0)])$.
dt, dtonrm	real	time step sizes for collision term and guiding-center motion

Note: In the mail loop of FORTEC-3D, almost all quantities are normalized. The normalization factors are determined by sb0=B0, sx0=minor radius, sv0=v_thermal at rho=1/3 flux surface, and pmass(proton

mass) or emass0(electron mass). All the other normalization factors are determined by the combination of above 4 quantities.

Phase-space meshes

Name	Type	Description
nrcell, nacell, nzcell	integer	numbers of (ρ, θ, ζ)-meshes (collision, Source/Sink, weight-average, flux, viscosity, etc.)
drcell, dtheta, dzeta, zetamax	real	width of the cells
nobx, noby	integer	velocity-space meshes in $(\xi, v/v_{th})$. (weight-averaging)
vmax	real	max v/v_th for the velocity-space
ndvx, ndvy, vmax_dist	integer	velocity-space meshes used in DFPLOT (observation purpose)
dvol1	real	volume of (nrcell, nacell, nzcell) cells.
dvol2,dvol3	real	volume of nrcell flux-annulus (dvol3 is normalized).
rn0, rn1	real	dvol*n (numbers of background plasma particles in cells)
b2ave , Zeff	real	$\langle B^2 \rangle$ and Zeff at the center of nrcell-meshes
Taujk	real	collision time between j and k species
th_jk	real	θ_{jk} parameter for Sugama collision operator (factor used for collisions between two particle species with different temperatur)
backvj, backtj, backnj, backljk	real	v_th, n ,T, ln_Lambda on each flux surface, each species
ic, mm	integer	marker index and number in (nrcell, nacell, nzcell) cells.
icr, mmr	integer	marker index and number in (nrcell) cells.
jxi,jyi,nbtt	integer	2D-reduced marker position in the 5D phase space cells jxi=(v_para-pitch * theta* zeta) dimension, jyi=(velocity*radial) dimension

B.1.2 Other variables contained in the modules

PHYSP

Name	Type	Description
ifelc	integer	switch for particle species to be solved

MOD_TMP

Name	Type	Description
nomp	integer	# of OMP threads
icntX, lazst, lazed, mxst, mxed	integer	for MPI work sharing and collective communications

TABLE_ERF

Table-data of collision frequencies for the linearized collision operator

Name	Type	Description
meshERF	integer	number of meshes in normalized velocity x=x_b/x_a
xmxtbl	real	maximum x value of the tables

EVOL_ER

Name	Type	Description
epp	real	$\epsilon_\perp$, permittivity (including the classical dipole effect)
EradX (X=0,1)	real	radial electric field (0: full-mesh, 1: half-mesh)
TVCav, PVCav, TVSav, PVSav	real	Fourier-decomposed toroidal and poloidal viscosity (cos-modes and sin-modes)
wflx	real	weight factor to fit Gamma between two flux surfaces

GAMMA_TABLE

Table-data of $\Gamma(\rho, E_r)$ read from file #19

CHECK_BAD

Name	Type	Description
kdameX	integer	Filter information
diffwsum_sum	real	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)

KM_RAND_WRAPPER

Variables and wrapper subroutines to use kmath_random routines.

B.1.3 Local variables in subroutines

VISC_FLUX_ER (evaluate neoclassical fluxes & viscosities)

Name	Type	Description
vir, qir, vpb	real	radial particle and energy flux, parallel flow UB
dni, dnT	real	particle and pressure of delta-f part
GamiX, GameX (X=0,1)	real	radial ion and electron particle fluxes

LINEARC (collision operator)

Name	Type	Description
sww, spz, see	real	change in the velocity-moments (particle number, parallel momentum, and energy) by collision term in each cell
AA, BB	real	3×3 matrix for the field-particle operator and the S/S term
WWWx,PPPx,EEEx	real	velocity moments in each cell

W_AVE

Name	Type	Description
aijX	real	3×3 matrix for the averaged weight field
bbx, ccx	real	3D vector of moments to be conserved by the weight averaging
WAW, WAP	real	fitting coefficients of the obtained averaged field

B.2 I/O units

I/O	File name	Description
7	outXXX_YYY.07	Radial profiles of E_r and Γ_d, Γ_{vnc} (radial flux evaluated from radial drift velocity v_dr and viscosity, respectively) at each time step, and Gam_tv, Gam_pv , which are fluxes proportional to NTV and NPV (see Satake, PPCF 2012). If a calculation has enough accuracy, Gamma_d and Gamma_vnc should coincide with. This file is created only if neoclassical viscosities are evaluated.
27	outXXX_YYY.07	Radial profiles of E_r and Γ_d, energy flux Q, $\langle U_\parallel B \rangle$, at each time step. Also contains dn, dnT, errors in f_0 moments, and the breakdown of radial flux by v_m and $E_1 \times B$ (non-zero only if phi1_effect is on).
14	out_YYY.14 (bin)	Spacial profiles of background plasma particle numbers in 3D cells rn0 and rn1. This file is created only when ngo=0, and is read for ngo≥ 1 jobs.
15,16,17	outXXX_YYY.15,16,17	Log files for the marker recycling (15), filter(16), and weight conservation(17) for each species. At the end of a run, these files are summed up to outXXX.(15,16,17).
8,9,30	outXXX_YYY.08,09,30 (bin)	Profiles of delta-f distribution function, average weights, and interpolated weight fields, respectively. This file is created only if ngraph/=0. To read these files requires a post-process code DFPLOT (§*.*).
41,42,43	out_YYY.41, 42, 43	Initial marker distribution and initial marker weight information. To plot, use "chk-markdist.gpl". (§7.2.8)
46	outXXX_YYY.46	Check log for the weight-averaging. To plot, use "ckh46.sh" (§7.2.3)
63,64,65,66	outXXX_YYY.63,64,65,66 (bin)	Time evolution of the neoclassical toroidal and poroidal viscosities (Sin- and Cos- modes). These files are created only if NTV and / or NPV evaluation switch is on. To read these files, use AV-FLVC. (§7.2.1)
98	out_XXX.98	Record of the parameters used in the XXX job.
26, 106	outXXX.26 outXXX_YYY.106	General information of the job XXX (26) and for species YYY (106). User can see the progress of job during it is running by watching this.
107,108	outXXX_YYY.107,108 (bin)	New format of Er, Gamma, Q, <UB>, <qB>, dn, dnT evolution data. These files are intended to reduce the time to read outXXX_YYY.07 and 27 text-style files in order to speed up the post-process code AV-FLVC. (§7.2.1)
127	outXXX_YYY.127 (bin)	Output file of transport coefficients D_{ab}^{kl} and driving forces X_b^l.

Output files related to Φ_1-effect :

140	outXXX_YYY.140	Filter log of $\Phi_1/\delta n$ calculation. Opened only if wwph1_log=.true.
141	outXXX.141	Φ_1 profile read at the beginning of a job (basically for debugging purpose).
142*	outXXX.142	Mapping data of Φ_1 and δn.
143*	outXXX.143	Mapping data of split δn. This file will be output only if wsplit> 0.
144	outXXX.144	Profile of $\Phi_0(r) = -\int^r E_r(r')dr'$ at the beginning of the job.
145	outXXX_phi1_output.145	Φ_1 and δn data for continuing the internal $\Phi_1/\delta n$ calculation. When being read as a continue file, its I/O number is 150.
146*	outXXX_YYY.146	Monitoring data of $\Phi_1/\delta n$ calculation.
147	outXXX.147	Monitoring data of $\Phi_1/\delta n$ calculation every ph1stp steps. Common for all species.
148*	outXXX.148	Spectrum data of Φ_1, δn, vd_sin, and vd_cos for plotting.
150	read	Reads a continue file of Φ_1 profile.
152	outXXX_YYY.152	Log file of energy conservation. Output only if E_consv=1,

Note: The file numbers with "*" contains information only for the last ph1stp steps averaged data.

Appendix C

Known problems

C.1 Noise and accuracy

- 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.
- Due to the code tuning and reducing the memory usage, the control-variate method developed by S. Matsuoka has been removed. 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.

C.2 Limitation of the model

- 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.

Appendix D

Help

D.1 Visualization

The graphs in the dev-XXX.ps appear to be strange.

- Adjust the paramer ntmp
- When the graphs appear to be like Fig. D.1, make sure that the parameter nr in the gnuplot file dev-erfl_XXX.gpl is the same as nrcell in the main program.

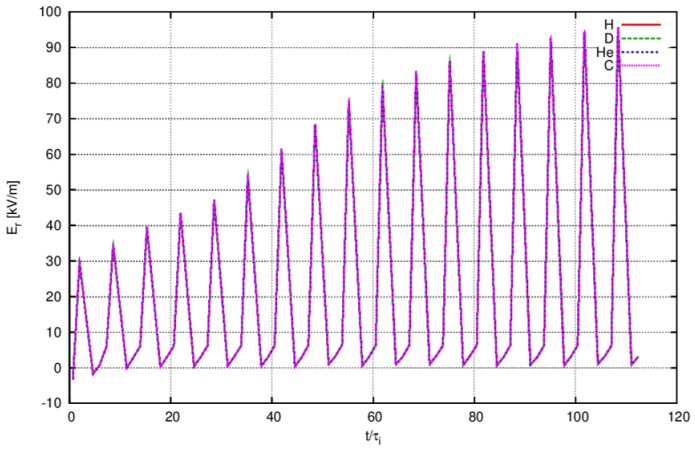

Figure D.1. Sample of the graph of the evolution of E_r generated with a wrong nr value: nr=50 in the gnuplot file, while nrcell=40 in the main program.

Appendix E

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 $\times$ 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
- "Effect of magnetic shear and the finite banana-orbit width on the neoclassical toroidal viscosity in perturbed tokamaks" S. Satake, I. Calvo, J. L. Velasco, M. Honda, S. Matsuoka. IAEA-FEC 2018 preprint, https://conferences.iaea.org/event/151/papers/6428/files/4581-iaea2018-preprints-satake_v4.pdf
- "Global Effects on the Variation of Ion Density and Electrostatic Potential on the Flux Surface in Helical Plasmas" K. FUJITA, S. SATAKE, R. KANNO, M. NUNAMI, M. NAKATA and J. M. GARCÍA-REGAÑA Plasma and Fusion Research Vol. 14, 3403102 (2019), https://doi.org/10.1585/pfr.14.3403102
- "Global calculation of neoclassical impurity transport including the variation of electrostatic potential" K. Fujita, S. Satake, R. Kanno, M. Nunami, M. Nakata, J. M. García-Regaña, J. L. Velasco and I. Calvo Journal of Plasma Phys. (2020), vol. 86, 905860319, https://doi.org/10.1017/S0022377820000598 "Benchmark of a new multi-ion-species collision operator for δf Monte Carlo neoclassical simulation" S. Satake, M. Nataka, T. Pianpanit, H. Sugama, M. Nunami, S. Matsuoka, S. Ishiguro, R. Kanno Computer Physics Communications vol. 255 (2020) 107249, https://doi.org/10.1016/j.cpc.2020.107249