NATIONAL INSTITUTE FOR FUSION SCIENCE

Documentation on DKES/PENTA with loose coupling with TASK3D-a

M. Sato and M. Yokoyama for TASK3D-UD (Users and Developers)

(Received - Jul. 21. 2016)

NIFS-MEMO-76 Aug. 31. 2016

TOKI, JAPAN

Documentation on DKES/PENTA with loose coupling with TASK3D-a

M. Sato and M. Yokoyama for TASK3D-UD (Users and Developers)

Abstract:
PENTA code [D. A. Spong et al., Phys. Plasmas **12** (2005) 056114, "PENTA_documantation.pdf" (provided privately from D. A. Spong), and others] calculates the neoclassical parallel flows, radial particle and energy fluxes, and the radial electric field for a surface given the plasma profiles (density and temperatures, including impurity ions), surface geometry information (from VMEC) and the mono-energetic transport coefficients (from DKES). A loose coupling to the integrated transport analysis suite, TASK3D-a [M.Yokoyama for TASK3D-Users and Developers, NIFS-Memo 61, Nov. 2012.] has been recently established in a sense that all the necessary files for PENTA (along with DKES) executions are automatically prepared based on the VMEC input (along with the plasma profiles) which is identified by the TASK3D-a execution for LHD discharges. In this documentation, how to execute each module towards DKES/PENTA, and furthermore, the semi-automated execution module of DKES/PENTA for facilitating the application of DKES/PENTA to LHD discharges are explained.

Keywords:
PENTA code, TASK3D-a, Plasma Simulator, LHD, neoclassical transport analyses

Table of Contents

1 OVERVIEW

Important modules are
- **VMEC,**
- **BOOZ_XFORM,**
- **DKES,**
- **PENTA,**

and the modules
- **PRO**,
- **PERP1**,
- **ECLIST**,
- **MAKE_PLASMA_PROFILES**,

are prepared as auxiliary modules for creating input data.

DKES and **PENTA** should be executed for each flux surface, and moreover, **DKES** has to be executed by varying radial electric field and collisionality. The following modules,

- **DKES_MPI**,
- **PENTA_MPI**,
- **PENTA_SQN**,

are prepared to control such numerous executions.

The overall calculation flow is shown in Fig. 1. The outline of this calculation flow is as follows.

1. Obtain the MHD equilibrium by **VMEC** (or import *input.vmec* for a designated timing/shot from TASK3D-a),
2. Create Plasma profile data, *profile_data_lhd*, by **PRO**. This file is required for **PERP1** and **PENTA**.
3. Create a data for the magnetic field spectrum, *boozmn.vmec*, by **BOOZ_XFORM** utilizing the VMEC-produced *wout.vmec*.
4. Create diffusion coefficient (such as L_{11}) database file, *tmp_xxxxx.txt*, by numerous **DKES** executions for a range of the radial electric field and the collisionality for each flux surface. Here *xxxxx* denotes the numbering such as 00001.
5. Create normalized diffusion coefficient (such as D_{11}^{*}) database file such as, *D11_star_lhd_sXXX*, by **PERP1** utilizing *tmp_xxxxx.txt* files as inputs. Here *XXX* denotes the radial mesh number.
6. Execute **PENTA** for each flux surface to evaluate neoclassical transport properties.

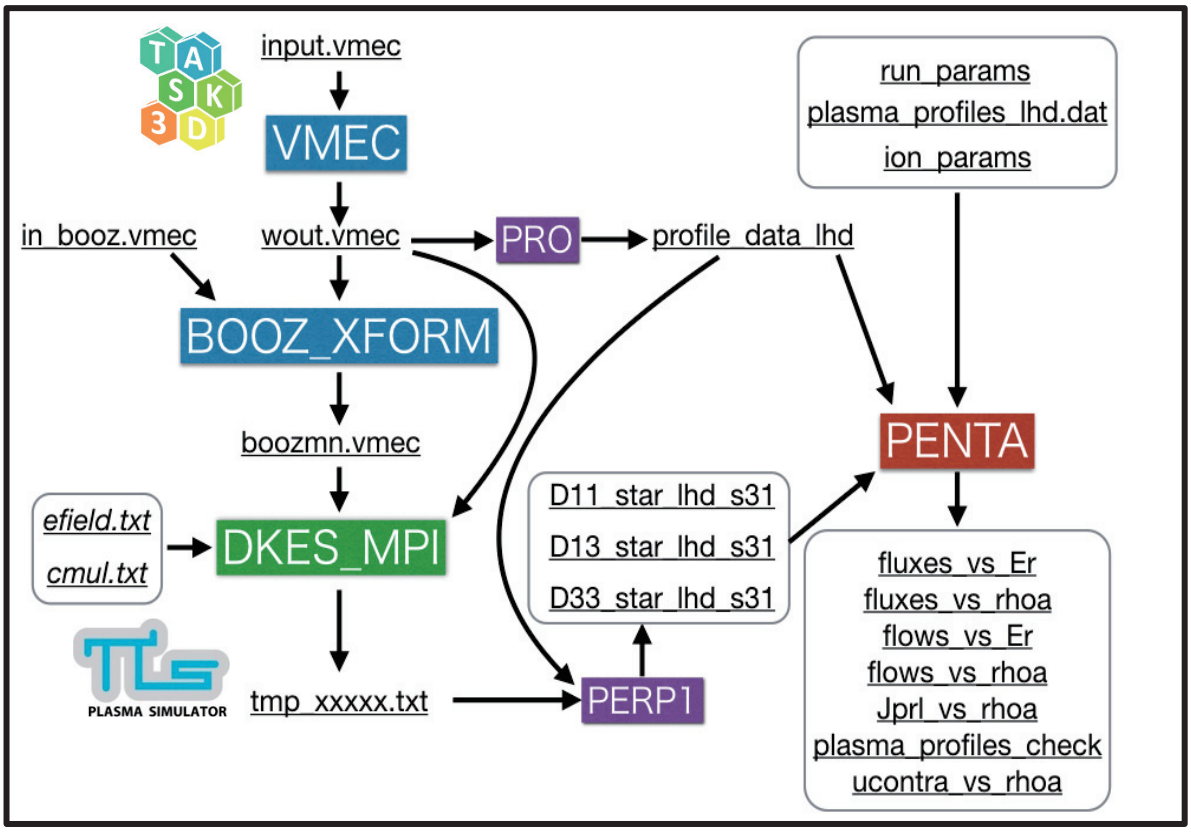

Figure 1 Overall calculation flow towards DKES/PENTA.

2 HOW TO EXECUTE EACH MODULE TOWARDS DKES/PENTA

2.1 VMEC

The file, *input.vmec*, is required as an input file. It can be created by your-own approach, or can be imported from **TASK3D-a server (on T:\\trsnaphd1.lhd.nifs.ac.jp\equilibrium at tsmap-task3d.lhd.nifs.ac.jp)**. The latter approach is recommended for analyses for (a) designated shot(s).

The shell script for **VMEC** execution is *go-vmec.csh* with the executable *xvmec2000*. The *wout.vmec* is one of output files, and it is required for successive modules.

2.2 BOOZ_XFORM

This program transfers the equilibrium information from VMEC coordinates to Boozer coordinates. The files,*wout.vmec* and *in_booz.vmec* are required as input files. Users must create *in_booz.vmec* by themselves as shown in Fig. 2. **BOOZ_XFORM** can be executed with *in_booz.vmec* as an argument.

```
./xbooz_xform in_booz.vmec
```

The *boozmn.vmec* is one of output files, and it is required for **DKES** execution.

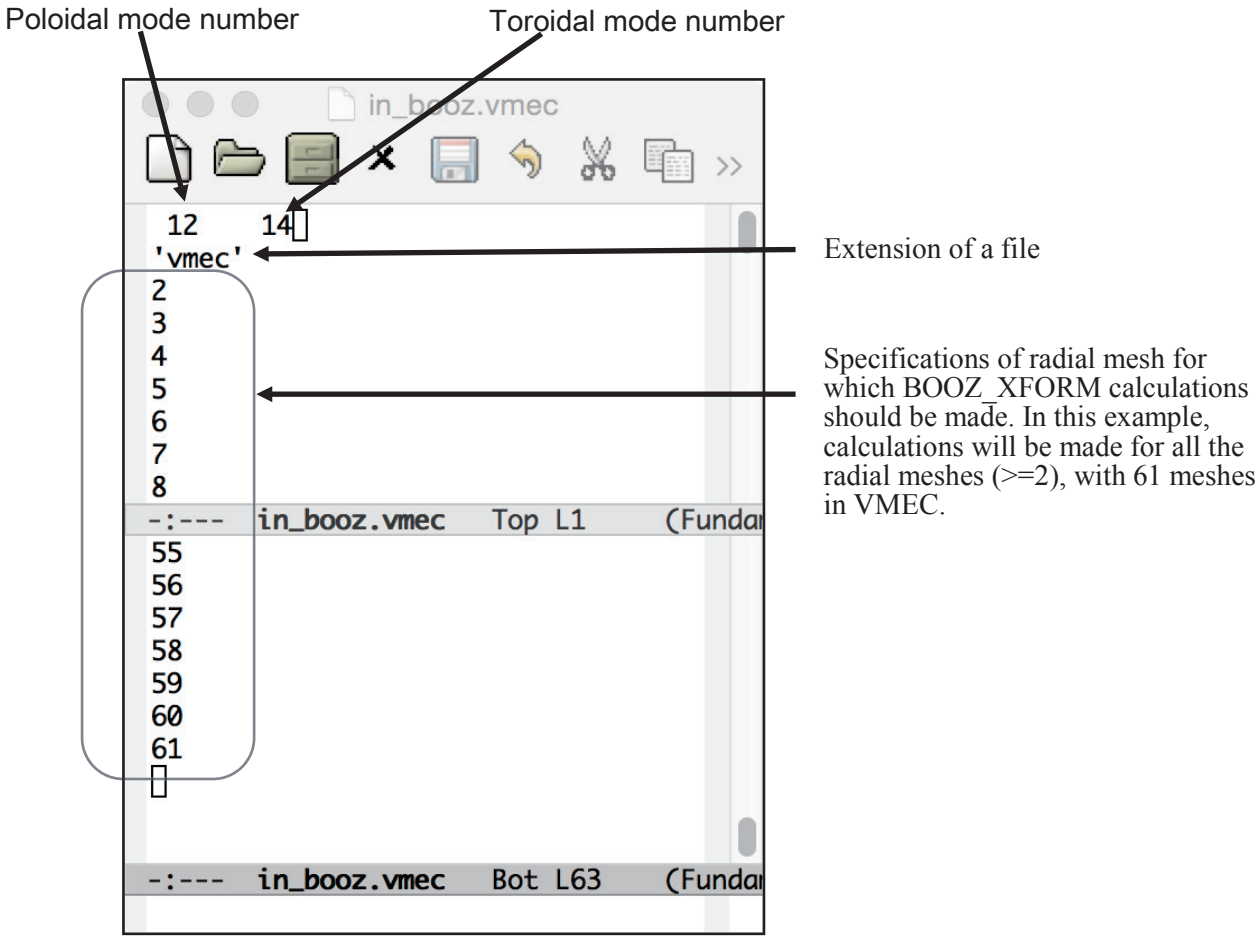

Figure 2 Contents of *in_booz.vmec*.

2.3 PRO

Profile information related to the equilibrium (such as χ'and ψ') are provided as *profile_data_lhd*,(as shown in Fig. 3) based on *wout.vmec*.

```
./xpro vmec
```

The file, *profile_data_lhd*, is required for **PERP1** and **PENTA** executions.

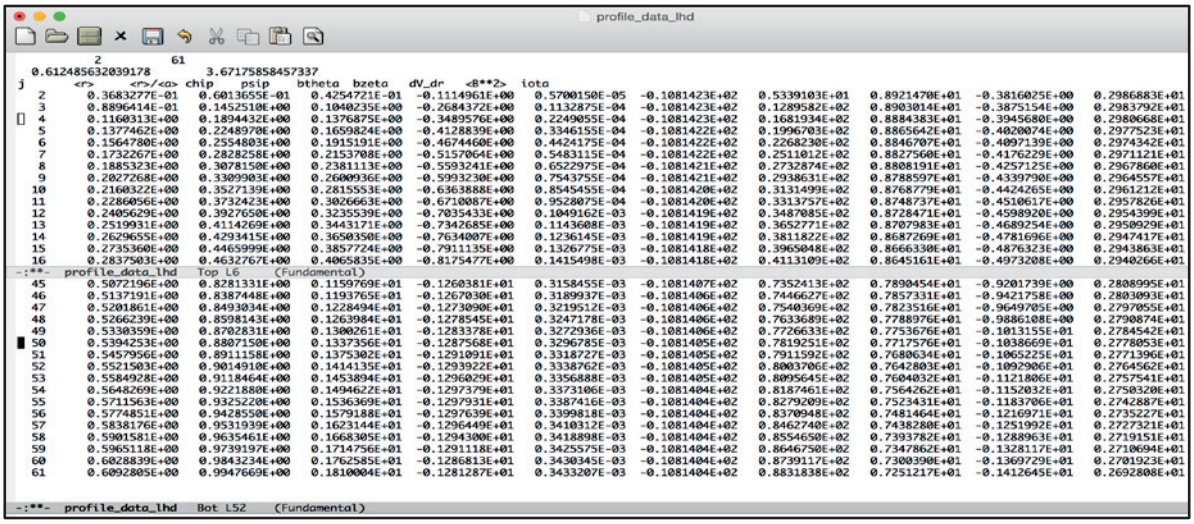

Figure 3 An example *profile_data_lhd*.

2.4 DKES

The two files, *wout.vmec* and *boozmn.vmec*, are required for **DKES** execution. Executions can be made as,

./xdkes boozmn.vmec 16 1.0e-2 0.0e-3 T 3 40

Meaning of each argument is as follows:

- (boozmn.vmec) File name of **BOOZ_XFORM**,
- (16) number of the radial mesh to be calculated,
- (1.0e-2) collision frequency (cmul: ν/v),
- (0.0e-3) radial electric field (efield: Er/v),
- (F) whether the plot on the screen in made (T) or not (F),
- (3) coupling order,
- (40) number of used Legendre modes.

Two files, *dkesout.vmec* and *results.vmec*, are created upon this execution. The *results.vmec* will be used Multiple **DKES** executions should be made with varying collision frequency and radial electric field on each flux surface, and then *results.vmec*s corresponding to those multiple runs are collected to create *tmp_xxxxx.txt*. Here, "*xxxxx*" designates the cases for radial electric field scan. For example, if radial electric field is varied as

1 0.0000000E+000,
2 1.0000000E-010,
3 5.8780161E-010,
4 3.4551073E-009,
5 2.0309176E-008,
6 1.1937766E-007,
7 7.0170383E-007,
8 4.1246264E-006
9 2.4244620E-005,
10 1.4251027E-004,
11 8.3767764E-004,
12 4.9238826E-003,
13 2.8942661E-002,
14 1.7012543E-001,
15 1.0000000E+000,

tmp_00001.txt (collected for cased with efield=0.0), *tmp_00002.txt* (collected for efield=1.0000000E-010), ,,,,, ,*tmp_00015.txt* (collected for efield=1.0000000E+000), should be created.

A *results.vmec* is organized as shown in Fig. 4 (the first line is only *, the second line is for variables and the third line is for values).

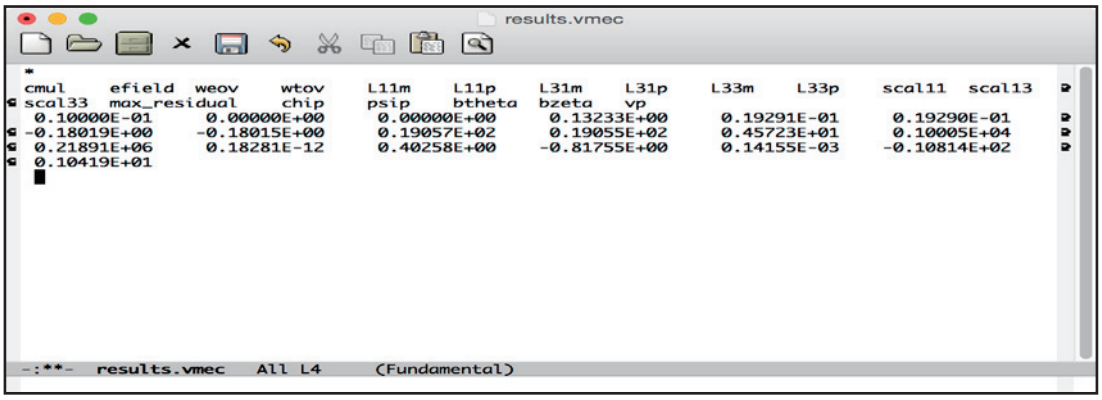

Figure 4 Contents of *results.vmec*.

tmp_xxxxx.txt is the file which gathers all the data in *results.vmec* on the descent order of the collisionality. *tmp_xxxxx.txt* is required to make files such as *D11_star_lhd_sXXX* as described later.

2.5 PERP1

This program is to read *tmp_xxxxx.txt* and then to create data files for diffusion coefficients such as D_{11}, D_{13}, D_{33} which are required for execution of **PENTA**. The required files for **PERP1** execution are *tmp_xxxxx.txt*, *cmul.txt* and *efield.txt*. The number of files is determined by the number of the parameter scans of efield. The files, *cmul.txt* and *efield.txt* store the parameters of cmul and efield, respectively, as shown in Fig. 5. The second line is for the number of parameter scans, and cmul and efield values are listed in the descent (cmul)/ascent (efield) order from the fourth line.

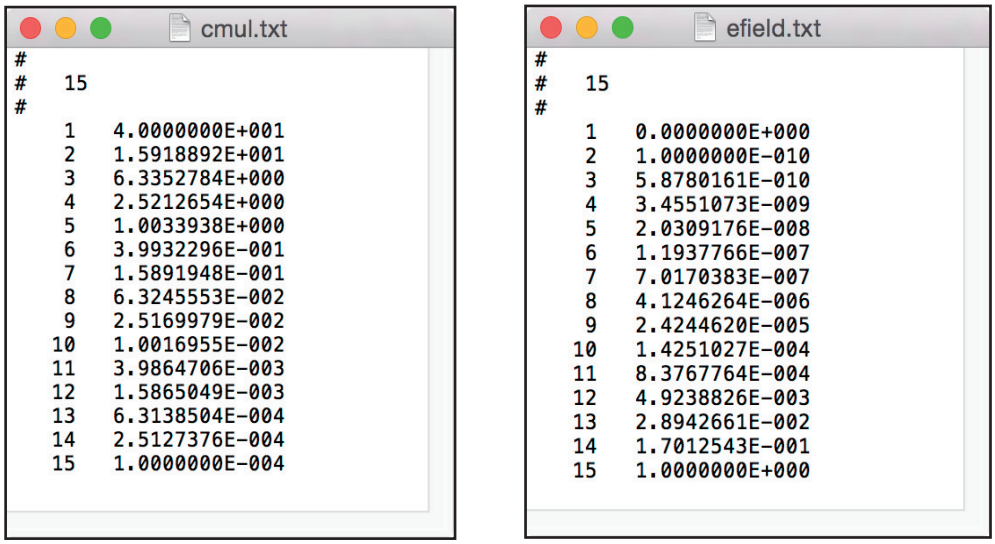

Figure 5 Contents of *cmul.txt* and *efield.txt*.

Executions can be made as,

./xperp1 lhd profile_data_lhd 10 10 1

Meaning of each argument is as follows:

- (lhd) the index to be added to the file name,
- (profile_data_lhd) the file name of the output from the program,
- (10) the radial mesh number from which the calculation is started,
- (10) the radial mesh number at which the calculation is stopped,
- (1) the interval of the mesh number for calculations (for example, calculations are done for the radial mesh numbers with the interval of 3, starting from the starting mesh number).

The above example execution makes the calculation for the radial mesh number of 10, which creates output files; *D11_star_lhd_s010*, *D13_star_lhd_s010* and *D33_star_lhd_s010*. For example, the contents of *D11_star_lhd_s010* is shown in Fig. 6.

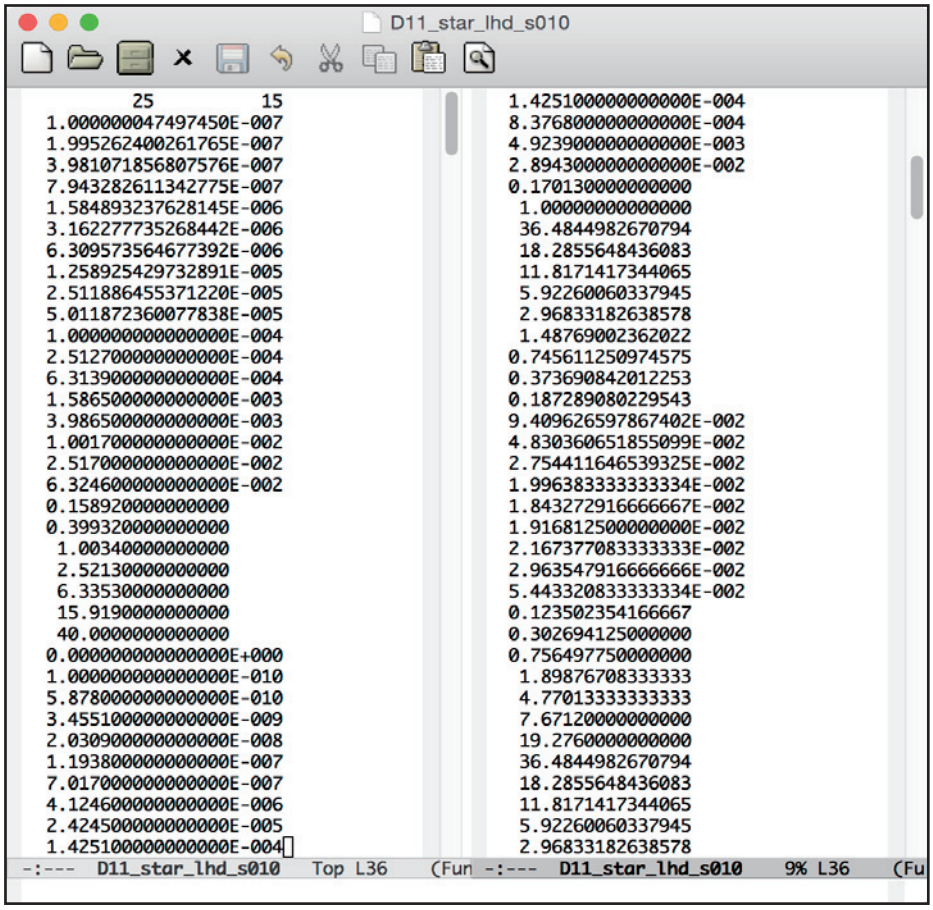

Figure 6 Contents of *D11_star_lhd_s010*.

The data in *D11_star_lhd_s010* can be retrieved by the following program. Here, nc and ne are the numbers of scanned cmul and efield, respectively, and the 2-dimensinal array D11(ic,ie) is for D_{11} data. .

```
============================
 read(iunit_d11,*) nc,ne

 do ic=1,nc
    read(iunit_d11,*) cmul(ic)
 end do

 do ie=1,ne
    read(iunit_d11,*) efield(ie)
 end do

 do ie=1,ne
    do ic=1,nc
      read(iunit_d11,*) D11(ic,ie)
    end do
  end do
============================
```

2.6 PENTA

This program is the last step of this overall calculation flow. The required input files are as follows:

- *run_params*

- *plasma_profiles_lhd.dat*
- *ion_params*
- *profile_data_lhd*
- *D11_star_lhd_sXXX* (*XXX*: radial mesh number)
- *D13_star_lhd_sXXX*
- *D33_star_lhd_sXXX*
- *Utilde2_profile* (optional)

The *profile_data_lhd* is the output file of **PRO**, and *Dxx_star_lhd_sXXX* files are the output files of **PERP1**. Thus, input files to be newly prepared here are; *run_params*, *plasma_profiles_lhd.dat* and *ion_params*. As for *Utilde2_profile*, please refer below.

Note)
Utilde2_profile - Contains the quantity $<U^2>$, where U is the Pfirsch-Schlüter flow function as defined by Sugama and Nishimura. The first row is the number of points, then r/a points and the corresponding $\langle U^2 \rangle$ value. Note that if read_U2_file is .false. , then $\langle U^2 \rangle$ is calculated from D_{11} at high collisionality.

The following parameters are defined in &run_params .

```
&run_params (details in penta.f90)
input_is_Er    = .true.          : If true, Er range is [V/cm] else e<a99>Er/kTe
log_interp     = .true.          : If true, logarithmic -interpolation of DKES coefficients is used
use_quanc8     = .false.         : If false, rect. approximation to convolution used
read_U2_file   = .false.         : If false <U^2> is calculated from D11*
Add_Spitzer_to_D33 = .true.      : If true collisional portion of D33* is added else it is assumed to be included in-file
num_Er_test = 50                 : Number of Er points in search range
numKsteps   = 10000              : Number of K (normalized kinetic energy) points (linear) for convolution
                                   (used if use_quanc8=.false.)
kord_pprof  = 3                  : Spline order for plasma profile fitting (and <U^2> file)
keord       = 2                  : Spline order for DKES coefficients fitting (for efield)
kcord       = 2                  : Spline order for DKES coefficients fitting (for cmul)
Kmin   = 1.d-5                   : Minimum K in energy convolution
Kmax   = 20.d0                   : Maximum K in energy convolution
epsabs = 1.d-8                   : Absolute tolerance for quanc8 (used if use_quanc8=.true.)
epsrel = 1.d-6                   : Relative tolerance for quanc8 (used if use_quanc8=.true.)
Method  = 'SN'                   : Which algorithm to use.  Options are
                                         'T'    = Taguchi
                                         'SN'   = Sugama-Nishimura
                                         'MBT'  = Maassberg-Beidler-Turkin
                                         'DKES' = Direct energy convolution
flux_cap=.true.                  : If true, min(Lradial) = 0
                                   (enforcing minimum radial diffusion coefficient is set to zero,
                                   for Method other than 'DKES', cf., penta.f90)
output_QoT_vs_Er = .false.       : If true Q/T vs Er output file is written
&end
```

The following parameters are defined in &ion_params.

```
&ion_params
num_ion_species=1   : number of ion species
z_ion_init=1.0d0,   : the corresponding ion charge numbers (real, array) for each non-electron species
miomp_init=1.0d0,   : the ion to proton mass ratio (real, array) for each non-electron species
&end
```

The *plasma_profiles_lhd.dat* is the file for density and temperature profiles, as shown in Fig, 7. The first line is the dimension number (arbitrary, not necessary to match with that of **VMEC**), and from the second line, normalized minor radius (r_{eff}/a_{99}), electron density (n_e), electron temperature (T_e), ion density (n_i) and

ion temperature (T_i) (information for ions to be repeated according to the number of ion species). The units for the densities and temperatures are [10^{18} m^{-3}] and [eV], respectively. It should be noted that the unit for the density is [10^{19} m^{-3}] in *params.txt*, but [10^{18} m^{-3}] is employed here in make_plasma_profiles.f90 which creates *plasma_profiles_lhd.dat*.

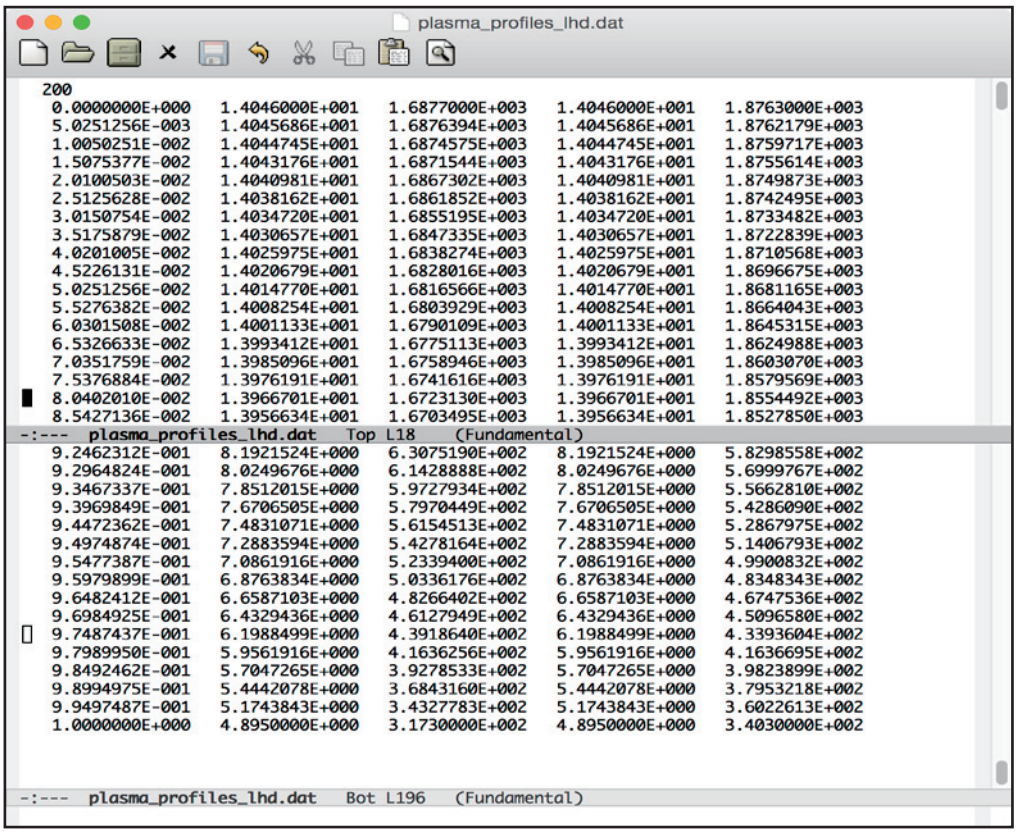

Figure 7 An example *plasma_profiles_lhd.dat.*

Executions can be made as,

./PENTA3 lhd_s010 -1.5e2 1.5e2 10 0 lhd lhd 0.0 2 .

Meaning of each argument is as follows:

- (lhd_s010) Index for the database files for D_{11}, D_{13} and D_{33}. In this case, *D11_star_lhd_s010* etc. are read,
- (-1.5e2) The minimum value of the radial electric field for searching the ambipolar condition [V/cm] (cf., see input_is_Er),
- (1.5e2) The maximum value of the radial electric field for searching the ambipolar condition [V/cm],
- (10) designation of the radial mesh number for the calculation,
- (0) The output file is (0) newly created, (1) overwritten to the existing file,
- (lhd) index (***) for the file, *profile_data_****,
- (lhd) index (***) for the file, *plasma_profiles_****.dat*,
- (0) the value of the electric field parallel to the magnetic field,
- (2) Smax --- The upper limit on the summation of Laguerre polynomial terms. Since the summation is from 0 to Smax, the number of terms used is Smax+1. This affects the number of parallel flow moments.

The output files are as follows (details in penta.f90);

- *flows_vs_Er* : Contains the Sonine weighted parallel flow moments for each Er in the ambipolar search range.

- *flows_vs_roa* : Contains the parallel flow moments for each species evaluated at the ambipolar radial electric field. The format is: r_{eff}/a_{99}, E_r, $ea_{99}E_r/kT_e$, $\langle B^* u_{\|ke}\rangle/\langle B^2\rangle$, $\langle B^* u_{\|ki}\rangle/\langle B^2\rangle$ ...

Here $\langle\ \rangle$ indicates a flux surface average, $u_{||}ks$ is the Sonine polynomial of order "k" weighted parallel flow moment for the species "s". So, after the E_r and $e<a_{99}>E_r/kT_e$ first each Sonine weighted parallel flow moment (from k=0 to k=Smax) is listed for the electrons, then each moment for the first ion species, each moment for the second ion species, etc.

- *fluxes_vs_Er* : Contains the radial particle fluxes for each surface for each value used in the ambipolar Er search loop. The format is: r_{eff}/a_{99}, E_r, $e<a_{99}>E_r/kT_e$, Γ_e, Γ_{i1}, Γ_{i2} , ...

- *fluxes_vs_roa* : Contains the radial particle and energy fluxes for each run surface for each ambipolar root. The particle fluxes Γ are listed for each species [units of particles/(m^{2}s)] and the energy fluxes as Q/T [units of 1/(m^{2}s)]. The format is given as r_{eff}/a_{99}, E_r, $e<a_{99}>E_r/kT_e$, Γ_e, Q_e/T_e, Γ_{i1}, Q_{i1}/T_{i1}, …

  Here the horizontal ... indicates that additional ion species will be listed horizontally, and the vertical ... indicates that additional ambipolar root results will be listed vertically. Each additional surface for a given run will then follow the same format and can be appended to this file.

- *Jprl_vs_roa* : Contains the parallel current densities in a similar manner as *fluxes_vs_roa* above. The format is r_{eff}/a_{99}, E_r, $e<a_{99}>E_r/kT_e$, $J_{prl,e}$, $J_{prl,i1}$, $J_{prl,i2}$, ... , J_{prl}.

  Here the units of the current densities are [A/m^2] and $J_{prl,s} = n_s q_s \langle b*u_{||s}\rangle$ where n_s is the density, q_s the charge and b is the normalized magnetic field $b=\langle B\rangle/\langle B^2\rangle^{1/2}$. The total parallel current J_{prl} is then the sum over species.

- *ucontra_vs_roa* : Contains the contravariant Boozer coordinate poloidal and toroidal components of the flow. These flows are given as $\langle u_{pol}\rangle = \langle \boldsymbol{u}\cdot\nabla\theta\rangle$ (toroidal case is similar). The units are thus [1/m]. The format of the file is r_{eff}/a_{99}, E_r, $e<a_{99}>E_r/kT_e$, $\langle u_{pol,e}\rangle$, $\langle u_{tor,e}\rangle$, $\langle u_{pol,i1}\rangle$, $\langle u_{tor,i1}\rangle$, ...

- *plasma_profiles_check* : Contains the plasma profile values actually used at each surface (T, n, and those gradients). Allows for the fits to the provided data points to be evaluated.

PENTA executions for each flux surface, and then merging each *fluxes_vs_roa* to one file provides the radial profiles such as of ambipolar radial electric field, particle and energy fluxes, and flow velocities.

3 SEMI-AUTOMATED EXECUTION MODULE OF DKES/PENTA

3.1 General remarks

The Ruby script (*run_vbdp.rb*) has been prepared to make the above described sequential calculations in a semi-automated manner (**VMEC**, **BOOZ_XFORM** and **PRO** on a local computer ⇒ **DKES** and **PERP1** on the Plasma Simulator ⇒ **PENTA** on a local computer). Here, how to utilize this script is explained.

(1) Prepare *input.vmec* and *params.txt*.
(2) Set parameters in *run_vbdp.rb*.
(3) Execute "ruby *run_vbdp.rb*".
(4) Copy files in directory FX (created by (3)) to the arbitrary directory on the Plasma Simulator.
(5) Execute "pjsub *go_dkes_fx.csh*" in the directory of (4) on the Plasma Simulator.
(6) When the job on the Plasma Simulator ends, copy files for D11 etc. in the directory PENTA (created by (5)) to the directory PENTA (created by (3)) on a local computer.
(7) Move to the directory PENTA on a local computer and then execute "*go_penta_mpi.csh*".

3.2 Input parameters

The contents of the input file, *params.txt*, to be prepared are as follows.

```
&param
#nr_min=16
#nr_max=16
#dnr=1
nr_list=16,30,40
mboz=12
nboz=14
ncmax=10
nemax=10
cmul_min=1.0e-4
cmul_max=4.0e1
efield_max=1.0e0
efield_min=1.0e-12
eps=2.0e-1
e_min=-200
e_max= 200
smax_penta=2
code_dir=/Users/masahiko/Dropbox/work/PENTA_release/ver2_test/bin input_vmec_file=input.vmec
input_booz_file=in_booz.vmec
coupling_order=3,4,5
legendre_modes=10,20,40,80,160
comp_dkes=fx
fx_class=X24
fx_node=12
fx_mpi_proc=384
fx_elapse=24:00:00
fx_account=16299
fx_code_dir=/data/lng/masahiko/TASK3D/dkes/bin
&end

&profile_param
num_ion_species=2
z_ion=1.0d0,6.0d0
mass_ion=1.0d0,12.0d0
cne =1.7126,-1.9779,4.1083,-1.582,-1.0552
cte =3.3056,-4.8812,11.895,-16.001,6.2889
cni(:,1)=1.6243,-1.9751,4.1034,-1.6989,-0.89641
cni(:,2)=0.014708,-0.00046668,0.00081616,0.01949,-0.026469
cti(:,1)=3.9691,-5.0715,2.7031,-0.95528
cti(:,2)=3.9691,-5.0715,2.7031,-0.95528
&end

&penta_param
input_is_Er    = .true.
log_interp     = .true.
use_quanc8     = .false.
read_U2_file   = .false.
Add_Spitzer_to_D33 = .true.
num_Er_test = 400
numKsteps   = 10000
```

```
kord_pprof  = 3
keord       = 2
kcord       = 2
Kmin   = 1.d-5
Kmax   = 20.d0
epsabs = 1.d-8
epsrel = 1.d-6
Method  = 'SN'
flux_cap=.true.
output_QoT_vs_Er = .false.
&end
```

Other parameters should also be defined in a run script: *run_vbdp.rb*, as follows.

- nr_min, nr_max, dnr, nr_list

These parameters define the range of radial meshes for which calculations are performed. Set nr_min, nr_max, dnr, or, set nr_list. Put # for unused options. (nr_min = 2, nr_max = 60, dnr=2) makes calculations for nr=2, 4, ⋯, 60. If nr_list = 16, 35, calculations are made for nr= 16, 35. Here nr should be larger than or equal to 2.

- mboz,nboz

These parameters define the mode numbers used for **BOOZ_XFORM**.

- ncmax, cmul_min, cmul_max

These parameters define the scan-range of cmul for DKES executions. The ncmax cmul values are calculated in the range from cmul_min to cmul_max, and then they are written in the file, *cmul.txt*. DKES reads those values of cmul from *cmul.txt*, and executes sequentially. Here, cmul values are calculated so that they are equal-interval in a logarithmic scale. It should be noted that diffusion coefficients for collisionality below cmul_min are given based on the extrapolation.

```
dcmul=(log(cmul_max)-log(cmul_min))/dble(ncmax-1)
do i=1,ncmax
cmul(ncmax-i+1)=exp(log(cmul_min)+dble(i-1)*dcmul)
end do
```

- nemax, efield_max

These parameters define the scan-range of efield for **DKES** executions. The nemax efield values are calculated in the range from 0 to efield_max, and then they are written in the file, *efield.txt*. DKES reads those values of efield from efield.txt, and executes sequentially. Here, efield values are calculated so that they are equal-interval in a logarithmic scale (details in eclist.f90).

```
efield(1)=0.0d0
defield=(log(efield_max)-log(efield_min))/dble(nemax-2)
do i=2,nemax
  efield(i)=exp(log(efield_min)+dble(i-2)*defield)
end do
```

- eps

This parameter sets the target value for the **DKES** convergence check. **DKES** executions give the upper and lower bounds of the mono-energetic diffusion coefficients (such as L_{11p} and L_{11m} for L_{11}). The convergence is checked by using the following values so that err can be smaller than the set value of eps.

```
dl11=(l11m-l11p)/(l11m+l11p)*2.0d0
dl31=(l31m-l31p)/(l31m+l31p)*2.0d0
dl33=(l33m-l33p)/(l33m+l33p)*2.0d0
err=max(dl11,dl31,dl33)
```

- coupling_order, legendre_modes

These parameters define mode numbers employed in **DKES** calculations. For example, coupling_order=[3,4,5] and legendre_modes=[10,20,40,80,160] are set, the convergence is checked, firstly by fixing coupling_order to 3 and increasing legendre_modes from 10, to 20, to, …, to 160. If this sequential examination does not reach the convergence (eps), then, coupling_order is increased to 4, and then to 5, with the maximum values of the set legendre_mode. It should be noted, however, that the total time required for this convergence check scales as $(\text{coupling\_order})^3 \times (\text{legendre\_mode})$, and thus it is not practical to lower the value of eps more than necessary.

- comp_dkes, fx_class, fx_node, fx_mpi_proc, fx_elapse, fx_account

If DKES calculations are performed in the Plasma Simulator, set as comp_dkes='fx', and set parameters fx_class, fx_node, fx_mpi_proc, fx_elapse and fx_account properly. Total number of **DKES** runs may become a large number (for example, radial mesh: 2~61, cmul: 15, and efield: 15 results in the total number of 13,500 (=60×15×15)), and these will be executed in parallel.

- e_min, e_max

PENTA is executed for the range of the radial electric field, from e_min to e_max ([V/cm]).

- smax_penta

This parameter defines the upper limit on the summation of Laguerre polynomial terms in **PENTA**.

- penta_mpi, penta_mpi_proc

PENTA is executed in mpi by setting penta_mpi='yes' with the execution module, *penta_mpi.exe*. The number of parallel executions should be defined by the parameter, penta_mpi_proc. If penta_mpi='no' is set, then **PENTA** is executed successively, with the execution module, *penta_sqn.exe*. (These are only available in the mpi-environment).

3.3 Calculation flow

The calculation flow for the semi-automated executions are as explained sequentially.

[1] On a local computer

1-1. Create the directory VBD for executing **VMEC**, **BOOZ_XFORM** and **PRO**.
1-2. Create the directory FX which stores files required for executions on the Plasma Simulator.
1-3. Create the directory PENTA for executing **PENTA**.
1-4. Execute **VMEC**.
1-5. Execute **BOOZ_XFORM**.
1-6. Execute *eclist.exe* to create parameter lists, *efield.txt* and *cmul.txt*.
1-7. Create files, *go_dkes.csh* and *input_dkes_mpi.txt*, required for executions on the Plasma Simulator, and then store them in the directory FX.
1-8. Execute PRO to create *profile_data_lhd*, and then copy it to the directories FX and PENTA.
1-9. Execute *make_plasma_profiles.exe* to create *plasma_profiles_lhd.dat*, and then store it in the directory PENTA.
1-10. Create input files for **PENTA**, *ion_params*, *run_params* and *input_penta.txt*, in the directory PENTA.
1-11. Create execution script for **PENTA**, *go_penta_mpi.csh* or *go_penta_sqn.csh*, in the directory PENTA.
1-12. Create directories PENTA_p and PENTA_m if check_bounds='yes', and copy or create necessary files there.

[2] On the Plasma Simulator

2-1. Create working directories for each flux surface. For example, create the directory s010 for thr flux surface with radial mesh number of 10.

2-2. Create working directories for each combination of parameters efield and cmul. For example, create directories from e001_c001 to e015_c020 (total number of 300) for the total numbers of parameter scans are 15 (for efield) and 20 (for cmul).
2-3. Allocate DKES execution script, *go_dkes.sh*, on all the working directories such as s010/c001_c001.
2-4. Make symbolic link to *boozmn.vmec* and *wout.vmec* on all the working directories such as s010/c001_c001.
2-5. Execute *go_dkes.csh* successively on all the working directories such as s010/c001_c001.
2-6. Create *tmp_xxxxx.txt* from results.vmec obtained from DKES parameter runs.
2-7. Execute *xperp1*to create the diffusion coefficient database such as *D11_star_lhd_s010*.
2-8. Create the directory PENTA, and then store the diffusion coefficient database such as *D11_star_lhd_s010* there.
2-9. Accumulate *tmp_xxxxx.txt* for each flux surface to DKES/*tmp_sXXX.txt*. Also, accumulate *tmp_visc_xxxxx.txt* for each flux surface to DKES/*tmp_visc_sXXX.txt*.
2-10. Delete working directories such as s010.

[3] **PENTA calculations on a local computer**
3-1. Create working directories for each flux surface. For example, create the directory s010 for the flux surface with radial mesh number of 10.
3-2. Move or copy files necessary for **PENTA** calculations to all the working directories.
3-3. Execute execution module, *PENTA3*, on each working directory.
3-4. Create *fluxes_vs_roa.txt* and *flows_vs_roa.txt* by reading output files, *fluxes_vs_roa* and *flows_vs_roa* created on each working directory.

3.4 Convergence check of DKES calculations

DKES gives upper and lower bound of the mono-energetic diffusion coefficients (for example, L_{11p} and L_{11m} for the actual solution, L_{11}). Increasing coupling_order and legendre_modes which are input parameters to **DKES** can narrow the gap between this upper and lower bound, and then to provide more accurate mono-energetic diffusion coefficients. As an example, Fig. 8 shows the dependence of L_{11p} and L_{11m} on these input parameters for cases with cmul of (a) 10^{-3} and (b) 10^{-4}.

The following parameters are defined in *run_vbdp.rb* for the convergence check of DKES calculations (cf., description in Sec. 3.2).

```
coupling_order=[3,4,5]
legendre_modes=[10,20,40,80,160]
eps=0.2
```

This combination of parameter settings execute **DKES,** firstly by fixing coupling_order to 3 and increasing legendre_modes from 10, to 20, … , to 160. If this sequential examination does not reach the convergence (eps), then, coupling_order is increased to 4, and then to 5, with the maximum values of the set legendre_mode. It should be noted, however, that the total time required for this convergence check scales as (coupling_order)^3×(legendre_mode), and thus it is not practical to lower the value of eps more than necessary (especially in low collisional regime).

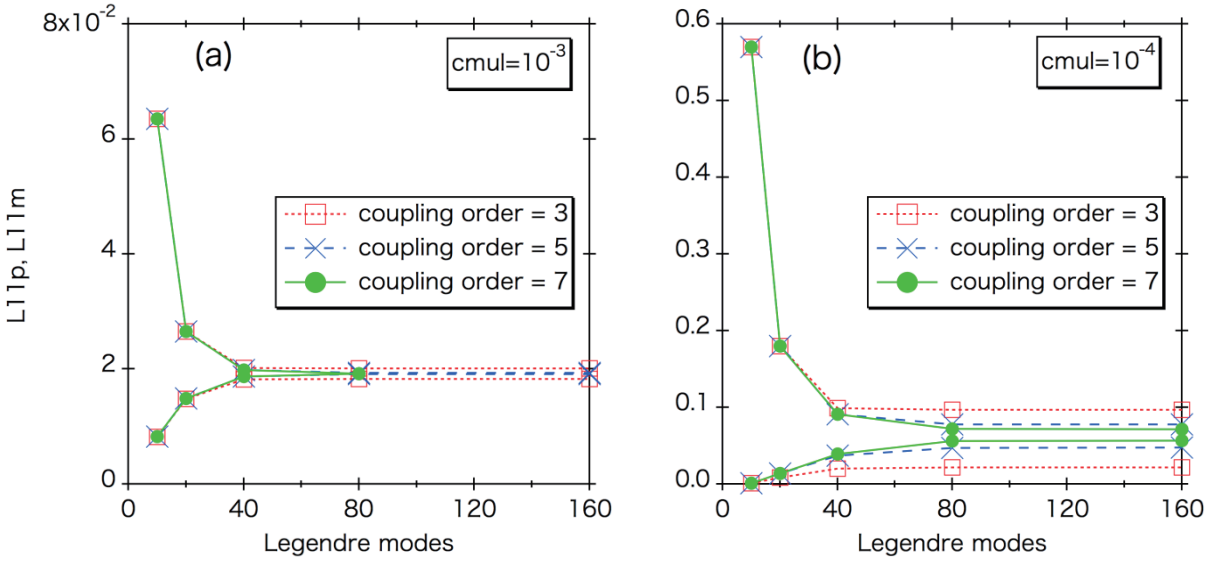

Figure 8 Dependence of L_{11p} and L_{11m} on parameters of coupling_order and legendre_mode for cases with cmul of (a) 10^{-3} and (b) 10^{-4}.

Acknowledgements:

The authors are highly grateful to Dr. D.A. Spong (Oak Ridge National Institute) for providing DKES/PENTA so that this loose coupling to TASK3D-a with semi-automated manner in NIFS can be conducted. This work is performed with the support and under the auspices of the NIFS Collaboration Research program, NIFS14KNTT025.