TASK3D-a Documentation

M.Yokoyama for task3d users-developers
Version a01: 2012-10-31

Table of Contents

1 Introduction

This documentation is for **task3d-a** (integrated transport analysis suite for LHD experiment). There is a mailing list for users and developers of task3d-a. If you want to join, please contact M.Yokoyama (yokoyama @ LHD.nifs.ac.jp).

Its first version, **task3d-a01**, was introduced at the LHD Experiment Group Meeting in 24 September 2012, and documented with this material.

The names of the contact person on each module (Sec. 3) are given for detailed consultation if required. It should be noted that those names are not necessarily those of developers.

2 TASK3D-a Usage

task3d-a can be executed in one of two ways:

1. Remote desktop connection to ***tsmap-task3d.lhd.nifs.ac.jp***
2. Mount the drive: ***T:¥¥trsnaphd1.lhd.nifs.ac.jp***
   (For execution of the program a windows PC must be used.)

ID: trsnap
password: trsnap

These machines can only be reached from ***within the LHD Experiment LAN***. When off-site, connections can be made by using a VPN connection.

2.1 Standard usage

- double click: pvwave.bat
- go, <shot#>

This will begin calculations using the standard settings, then automatically register the results with the eg system.

The contents of go.pro should NOT be modified.

The options explained below should NOT be used with the “go” command; using the default settings ensures that the calculation results registered with in ***eg*** use the same settings regardless of who initiates the task3d-a suite.

2.2 Personalized usage

- create your own directory at T: using your name (e.g. yokoyama).
- copy go-noreg.pro and pvwave.bat to your own directory
- double click: pvwave.bat
- go-noreg, <shot#>

go-noreg.pro does not perform eg-registration. The results including eg-format files are written out under your own directory. Thus, the contents of go-noreg.pro can be modified for your own purpose. The options explained below can be used.

2.3 Available options for personalized use

time=[time1, time2, time3,,,,,,]

(seconds)

Calculations will be done only at the specified time(s).

This will **override** the default timing settings.

(ex: time=[4.5, 4.633, 4.6, 5.8] → only these timings are calculated)

If this option is not set, calculations will be done for every time where the strong Thomson lasers are active.

add_time=[time1, time2, time3,,,,,,]

(seconds)

Time(s) given will be **added** to the default calculation times (times with strong Thompson lasers)

(ex: add_time=[4.5, 4.633, 4.6, 5.8] → default timings + add_time)

If both the “time” and “add_time” options are set, then all then given times will be used (no defaults).

cxs

Ti profile from cxs7 is prepared as ti_<#shot>t<time>.txt, and it is used for fit3d calculations. Otherwise, Ti=Te is employed for fit3d calculations. See, explanations for fit3d module.

delay=[time]

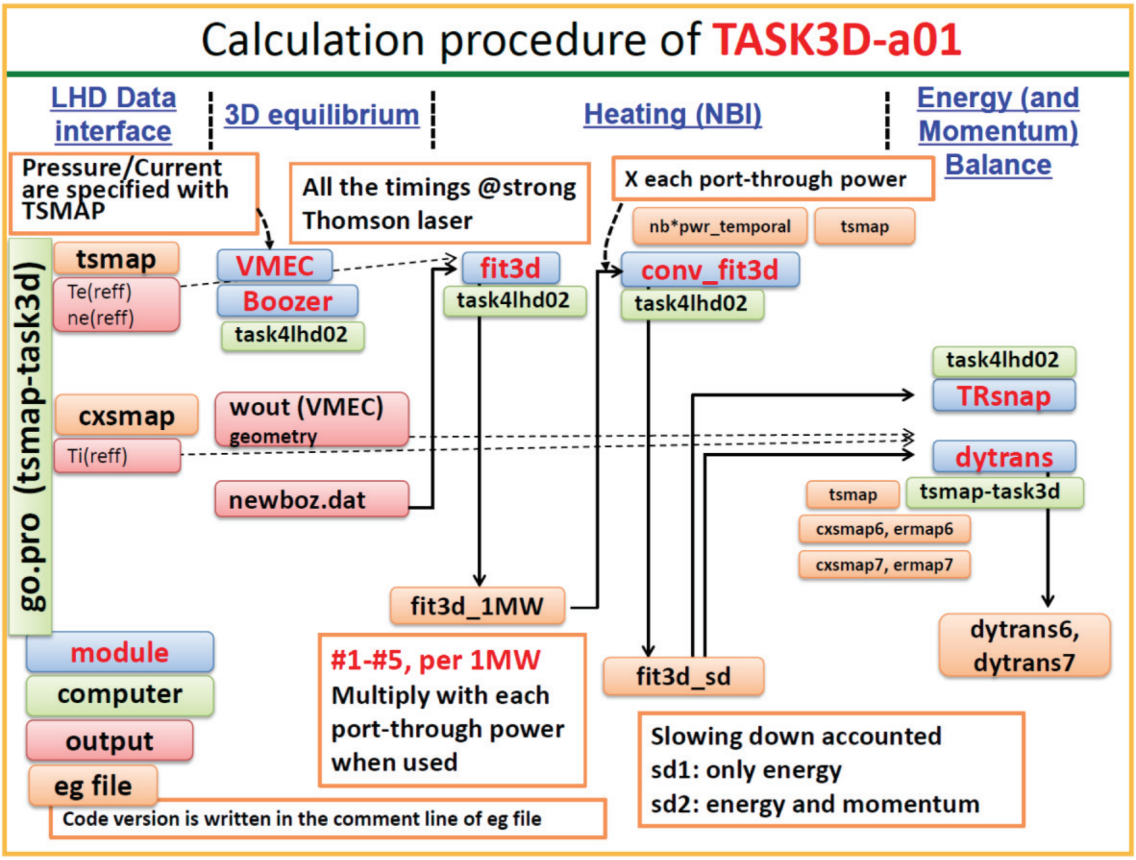
(seconds)

Specifies the duration for the task3d-a calculations to be performed after turning-off the NBI. This only has an effect if default timings are used (or add_time).

Default value is 1.0 s.

Brief description for each module is given in Section 3 with the name of contact person.

Figure 1



3 Description on modules employed in task3d-a

3.1 VMEC

Y.Suzuki

VMEC2000_6.90 is employed in task3d-a01. It calculates MHD equilibrium (fixed boundary calculation). The VMEC equilibrium database (VMEC-DB) for TSMAP has been prepared with VMEC2000_8.0 by Y.Suzuki. Details for TSMAP can be found in Ref. [1]. The equilibrium solution used for each requested time slice is re-calculated by utilizing parameters of so-called “best-fit” TSMAP.

3.1.1 Input

Inputs used for VMEC calculation:

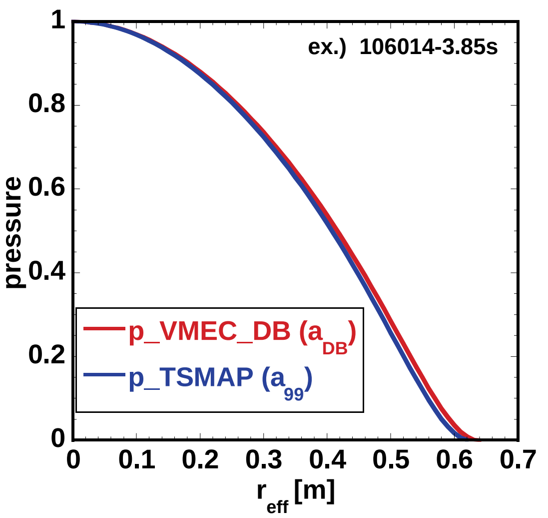
- Last closed flux surface (R_{mn} and Z_{mn}): taken from VMEC-DB corresponding to a_{99}. Here, a_{99} is the minor radius in which 99% of the total stored energy is confined.
- Pressure profile: p_0 (peak beta value) and p_f (pressure peaking factor) are taken from “best-fit” TSMAP. These 2 values provide the functional form of pressure (“am” in the namelist). If a_{99} differs from the one corresponding to the minor radius in the VMEC-DB (say, a_{DB}), $\psi = (r/a_{99})^2$ is approximately used, instead of $\psi=(r/a_{DB})^2$.
- Current profile: “curtor” is provided with ***eg: ip***. As for a profile, it is assumed to be proportional to $1-\psi^2$.

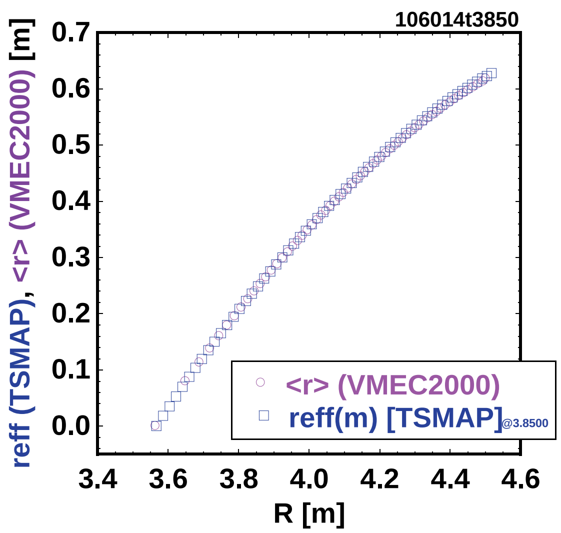
- “phiedge” is provided by phiedge0*$(a_{99}/a_{DB})^2$, where phiedge0 is the toroidal flux for a vacuum case in VMEC-DB. If phiedge0 and a_{DB} do not exist in VMEC-DB (such as for R_{ax}=3.53m), interpolation is performed with available data.
- The default setting for calculation conditions is as follows
  ns=61, nu=32, nv=32, mu=8, mv=6, (u = theta, v = zeta), ftol=1.e-15

The VMEC input file can be found in the directory:
    T:¥equilibrium¥XXXXXX¥XXXXXXtYYYY

Since the “best-fit” TSMAP is defined by just satisfying “Thomson peak (Te,max) to the magnetic axis” and “in-out symmetry”, it does not necessarily mean all of the equilibrium properties are well reproduced. Thus, it should be considered that this approach is just one of practical approaches for providing equilibrium for experimental analysis. The “wout” file (produced by VMEC) can be replaced by that based on other equilibrium reconstruction approach to see/investigate the impact of equilibrium on the analyses.

Below, an example is shown on the comparison between the “best-fit” TSMAP (a_{99}~0.62m) and VMEC-DB (a_{DB}~0.63m) for a particular shot-timing. p_TSMAP is given to the input of VMEC2000. The approximation, $\psi=(r/a_{99})^2 \sim \psi=(r/a_{DB})^2$, provides little impact on the mapping between r_{eff} (TSMAP), <r>(VMEC calc.) and R.





3.1.2 Output

The VMEC output files can be found in the directory:
    T:¥equilibrum¥XXXXXX¥XXXXXXtYYYY

wout: standard output generated from VMEC2000. The variables written out in this file depends on the version, and it is currently ver6.90.

threed1: fundamental information (calculation evolution towards convergence, some equilibrium quantities etc.) are available.

dia08: it was a standard output in old versions of VMEC. It is also written out to be usable to some codes to read dia8 in previous days.

3.2 Boozer

M. Yokoyama

newboz performs the mapping from VMEC coordinated to Boozer coordinates, so that equilibrium is described in Boozer coordinates.

3.2.1 Input

The input is the VMEC wout file.
The default settings for mapping are as follows:

m_nbz = 64
nmax_nbz = 20
nmin_nbz = -20
ltsmap = .t.
lhs_to_rhs = .f.
nit_to_pit = .f.
vel_to_hel = .f.

- ***ltsmap*** was introduced to be compatible with VMEC-DB, where phiedge is given by negative value.

```
(in transform.f90)
!MY----- TSMAP -----
    if( ltsmap ) then
!MY -- ltsmap
! reversal of only wjs and wis (from original psi<0 -> psi>0 part)
            do i = 2, ns
            wjs(i) = -wjs(i)
            wis(i) = -wis(i)
            end do
! reversal of xiota, wjs, xnboz and pbozh (same as original iota<0 -> iota>0
part)
      do i = 2, ns
        xiota(i) = - xiota(i)
        wjs  (i) = - wjs  (i)
      end do
      do m = 1, nmboz
        xnboz(m) = - xnboz(m)
        do i = 2, ns
          pbozh(m,i) = - pbozh(m,i)
        end do
      end do
     print 1000
    endif
1000 format(/1x,' --------------------------------------'    &
            /1x,'   transformation (TSMAP) done          '    &
         /1x,' --------------------------------------'/)
     return
     endif
!MY---- TSMAP -----
```

3.2.2 Output

The newboz output files can be found in the directory:
T:¥equilibrum¥XXXXXX¥XXXXXXtYYYY

- **newboz_a_XXXXXXtYYYY.dat**: ascii
- **newboz_b_XXXXXXtYYYY.dat**: binary
- **newboz_XXXXXXtYYYY.txt**: accuracy of the mapping can be checked by columns under "check of relation between VMEC and Boozer", where the maximum value of the relative difference of B,R,Z,ϕ between VMEC and Boozer coordinates on each flux surface is written out.

3.3 fit3d [2]

S.Murakami (Kyoto University), R.Seki

fit3d has been developed ("reduced" version of GNET) to evaluate radial profiles of NBI absorbed power, beam pressure, beam source and induced momentum. The calculation consists of three parts as follows.

- HFREYA: calculations of the birth profile (from the generation of the beam particles in the beam source to ionization in the plasma)
- MCNBI: birth-ions are followed (shorter than the energy slowing-down time, but longer than the orbit effects such as prompt loss can be reflected)
- Steady-state solution of Fokker-Plank equation is obtained analytically without orbit effects taken into account

Required inputs

- Density and Temperature profiles: automatically created from afit_nT_coeff_XXXXXXtYYYY.txt (T:\equilibrium/XXXXXX/XXXXXXtYYYY). The polynomial form, **Y=c0+c1*ra^2+c2*ra^4+c3*ra^6** (here ra≡r_{eff}/a_{99}), is employed for the fitting. The illustrative comparison with the standard deviation between measured data and the fitting curve is collectively found in afit_summary_XXXXXX.pdf (T:\equilibrium/XXXXXX/XXXXXXtYYYY) for all the timings. There, the standard deviation for fitting are written out, and "Te(or ne)<0 points" is remarked if the fitting raises temperature and/or density values <0.
  3D equilibrium (Boozer): made by boozer module: specified by "newboz_file".

NB.)

- Ti is not always measured for selected timings by default in task3d-a01 (@strong Thomson lasers). Thus, Ti=Te is assumed for a standard use of task3d-a01 (based on known rather weak impacts of Ti on the deposition properties)
- eg file is written by assuming Port-through power to be1 MW for each beam line

3.3.1 Input

Cf., Appendix
[a part in input.trsnap_XXXXXXtYYYY.txt (T:\trsnap/XXXXXX/XXXXXXtYYYY)]

3.3.2 Output

eg: fit3d_1MW
DimNo=3
time [s]

# of beam line
reff/a99
ValNo=17

P_all [MW/m^3]:	total absorbed power
P_e [MW/m^3]:	absorbed power to electrons
P_i [MW/m^3]:	absorbed power to ions
Q_all [W]:	volume-integrated total absorbed power
Q_e [W]:	volume-integrated absorbed power to electrons
Q_i [W]:	volume-integrated absorbed power to ions
B_com [cm^{-3}]:	beam component density
com_per [Jcm^{-3}]:	perpendicular pressure of beam component
com_par [Jcm^{-3}]:	parallel pressure of beam component
en_perp [J]:	perpendicular energy of beam component
en_para [J]:	parallel energy of beam component
cur_dens [Acm^{-3}]:	beam current density
current [A]:	volume-integrated beam current
S [1/(s cm^3)]:	beam source density
total_S [1/s]:	volume-integrated beam source
mom_e [kgm/s^{2}cm^{-3}=Ncm^{-3}]:	momentum input density to electrons
mom_i [kgm/s^{2}cm^{-3}=Ncm^{-3}]:	momentum input density to ions

3.4 conv_fit3d

M. Osakabe, H.Lee

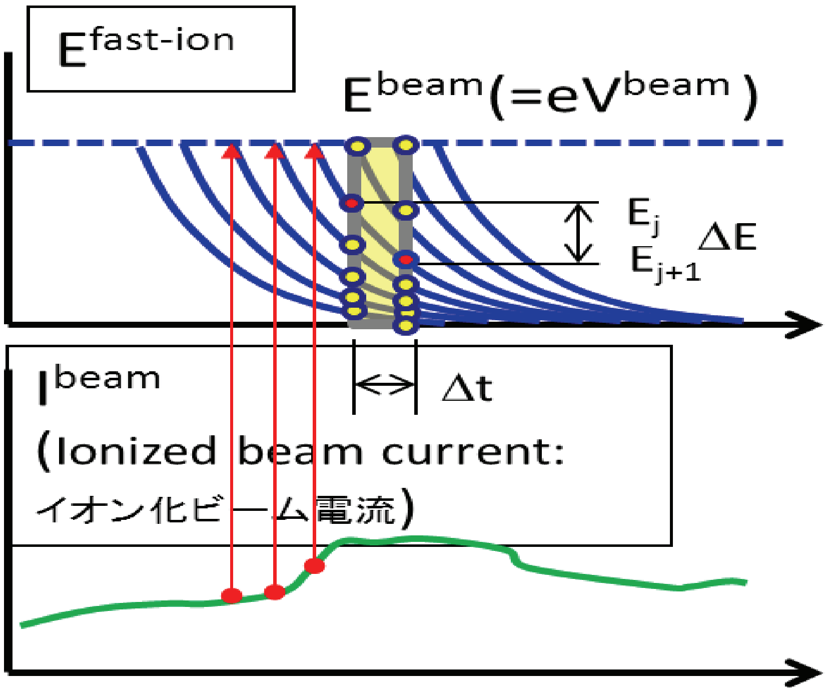
conv_fit3d has been developed to evaluate the NBI absorbed power and induced momentum by taking the beam slowing down (SD) effect into account, based on ***eg***: ***fit3d_1MW*** (which does not include SD).

- It is assumed that the ions with the injection energy (E_{inj}) are produced with typically Δt=100ms (corresponding to the interval of selected timings at fit3d calculations) during NBI injection. Those ions are followed until their energy becomes zero. Heating power is evaluated with considering ions with energy above Ti(=Te).
- Slowing down process for ions with E_{beam} at each timing is evaluated.
- Suffix j indicates that ions were injected at “j-th” previous time-steps from t_i. Thus, the ion energy injected at t=t_i is expressed by $E_{i,0}$(=E_{beam}). In a similar way, ion energy expressed by $E_{i,j}$ at t=t_i is denoted by $E_{i+1,j+1}$ at the next timing, t=t_{j+1}.
- Relationship between $E_{i,j}$ at t=t_i and its energy at the next time-step, $E_{i+1,j+1}$ at t=t_{j+1} is deduced from

$$E_{i+1,j+1} = \left[{E_{i,j}}^{3/2} exp\left(-\frac{3\Delta t}{\tau_{se}}\right) - {E_c}^{3/2}\left\{1 - exp\left(-\frac{3\Delta t}{\tau_{se}}\right)\right\}\right]^{2/3}$$

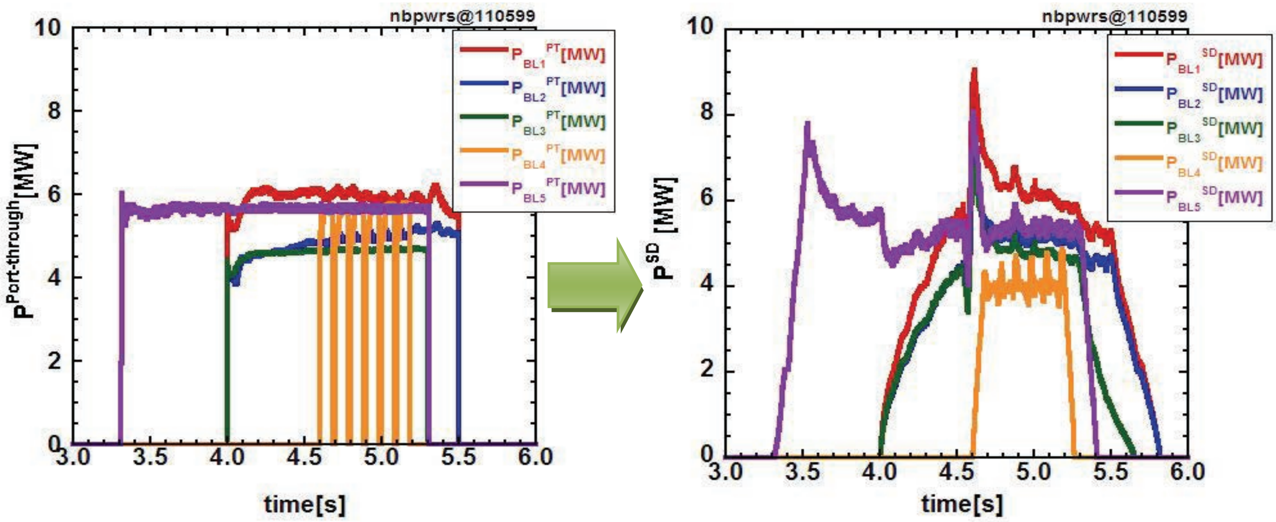
- Heating power within 1 time-step is calculated by the sum of $\Delta E_{i,j}=E_{i,j} - E_{i+1,j+1}$ by weighting the ionized beam current, $I^{beam}_{i,j}$.

This evaluation process is schematically shown below.



A particular example (calculated for #110599, pellet@~4.55s) is shown below.

- Temporal change of the density and temperature is taken into account (through dataset of sequential fit3d_1MW).
- Energy slowing down time is evaluated with the density and temperature @r_{eff}/a_{99}=0.5.
- SD-Momentum is evaluated in a similar way.



3.4.1 Input

Sequential series (typically with Δt=100ms) of ***eg: fit3d_1MW*** ,and
eg: tsmap, nb1pwr_temporal, nb2pwr_temporal, nb3pwr_temporal, nb4apwr_temporal, nb4bpwr_temporal, nb5apwr_temporal, nb5bpwr_temporal

3.4.2 Output

eg: fit3d_sd (slowing down)
DimNo=3
time [s]

# of beam line (1-5)	
reff/a99	
ValNo=23	
Pfit3d [MW/m^3]:	total absorbed power (***SD is not taken into account***) [=(P_all in eg:fit3d_1MW)*(Port-through power)]
P_all [MW/m^3]:	total absorbed power with ***SD taken into account***
P_e [MW/m^3]:	absorbed power to electrons with SD taken into account
P_i [MW/m^3]:	absorbed power to ions with SD taken into account
Q_all [W]:	volume-integrated total absorbed power
Q_e [W]:	volume-integrated absorbed power to electrons
Q_i [W]:	volume-integrated absorbed power to ions
Cf_tot:	correction factor for the total absorbed power [Total absorbed power: SD/no-SD]
Cf_e:	correction factor for the absorbed power to electrons [Absorbed power to electrons: SD/no-SD]
Cf_i:	correction factor for the absorbed power to ions [Absorbed power to ions: SD/no-SD]

B_com [cm^{-3}]:	beam component density (*as it is in eg:fit3d_1MW*)
com_per [J/cm^3]:	perpendicular pressure of beam component (*as it is in eg:fit3d_1MW*)
com_par [J/cm^3]:	parallel pressure of beam component (*as it is in eg:fit3d_1MW*)
en_perp [J]:	perpendicular energy of beam component (*as it is in eg:fit3d_1MW*)
en_para [J]:	parallel energy of beam component (*as it is in eg:fit3d_1MW*)
cur_dens [A/cm^3]:	beam current density (*as it is in eg:fit3d_1MW*)
Current[A] :	volume-integrated beam current (*as it is in eg:fit3d_1MW*)
S [1/(s cm^3)]:	beam source density (*as it is in eg:fit3d_1MW*)
total_S [1/s]:	volume-integrated beam source (*as it is in eg:fit3d_1MW*)

mom_SS [kg/m^{2}s^2=N/m^3]:	momentum input without SD taken into account = ((mom_e+mom_i) in eg:fit3d_1MW) * (Port-through power)]
mom_SD [kg/m^{2}s^2=N/m^3]:	momentum input with SD taken into account
F_SS [N]:	volume-integrated momentum input (without SD taken into account)
F_SD [N]:	volume-integrated momentum input (with SD taken into account)

3.5 TRsnap [3]

R.Seki, original: TASK/TR (A.Fukuyama, Kyoto Univiersity)

$$\cancel{\frac{1}{V'}\frac{\partial (n_j V')}{\partial t}} = -\frac{1}{V'}\frac{\partial}{\partial \rho}(V'\Gamma_j) + S_j$$

$$\cancel{\frac{1}{V'^{5/3}}\frac{\partial}{\partial t}\frac{3}{2}\left(n_j T_j V'^{5/3}\right)} = -\frac{1}{V'}\frac{\partial}{\partial \rho}(V' Q_j) + P_j \qquad V' = dV/d\rho, \quad \rho \equiv r_{\mathrm{eff}}/a_{99}$$

$$Q_j = -\left\langle |\nabla\rho|^2 \right\rangle \chi_j n_j \frac{\partial T_j}{\partial \rho} \cancel{+ \langle |\nabla\rho| \rangle u_j n_j T_j + \frac{3}{2}\Gamma_j T_j}$$

$$\Gamma_j = -\left\langle |\nabla\rho|^2 \right\rangle D_j \frac{\partial n_j}{\partial \rho} + \langle |\nabla\rho| \rangle u_j n_j$$

$$P_e = -\frac{n_e T_e}{\tau_{ei}} + \frac{n_i T_i}{\tau_{ie}} + P_{in,e} = P_{ie} + P_{in,e}$$

$$P_i = \frac{n_e T_e}{\tau_{ei}} - \frac{n_i T_i}{\tau_{ie}} + P_{in,i} = P_{ei} + P_{in,i}$$

$$\chi_j = -\frac{\int P_j \rho dV' \cancel{- \langle |\nabla\rho| \rangle u_j n_j T_j - \frac{3}{2}\Gamma_j T_j}}{\left\langle |\nabla\rho|^2 \right\rangle n_j \frac{\partial T_j}{\partial \rho}}$$

TRsnap has been modified based on TASK/TR (A.Fukuyama) to evaluate steady-state energy balance.

NB.) currently (in task3d-a01),

- Pin,e(i) is evaluated just from FIT3D (NBI). ECH and ICH have not been available. Other losses (=negative contribution, like radiation loss) have not been included.
- For χ_j, only the first term of denominator is considered.
- 3/2 in energy-transfer term.

3.5.1 Input

[a part in input.trsnap_XXXXXXtYYYY.txt (T:\trsnap/XXXXXX/XXXXXXtYYYY)]
&fit3dparam
ipg=0: *ion species (0: H, 1: He, 2: Ne)*
npart=103000: *default is 8000, varied for extreme low density cases (like EGAM shots) so as to be npart~ 1.28*10^{23}/n_0[m^{-3}]-43000 (based on advice from R.Seki)*
cn0=1.0e16: *edge neutral density: n0=cn0*exp*[*-(1-ra)/0.1*], ra=reff/a99
newboz_comp='local'
newboz_file='T:\equilibrium\109027\109027t5800\newboz_b_109027t5800.dat'
inbi1='on'
enbi1= 0.0000, pnbi1= 1.0000
enbi : read from **eg: nb*pwr_temporal** (zero for no injection)*
pnbi : all set for 1 MW (for **eg: fit3d_1MW**)*
inbi2='on'
enbi2= 0.0000, pnbi2= 1.0000
inbi3='on'
enbi3=176.4790, pnbi3= 1.0000
inbi4='on'

3.5.2 Output

The results are written out at the directory: T:\trsnap/XXXXXX/XXXXXXtYYYY with the suffix as follows (with/without SD)

- **_SS** : deposition power without SD consideration (eg:fit3d_1MW x Pport-through for each beam line) is used for energy balance.
- **_sd** : deposition power with SD consideration (eg:fit3d_sd) is used for energy balance.

<u>tr_snap_XXXXXXtYYYY.113</u> (T:\trsnap/XXXXXX/XXXXXXtYYYY)
NR : radial mesh number
rho: reff/a99
chi_e_exp [m^2/s]: χ_e
chi_i_exp [m^2/s]: χ_i
ne [10^{20} m^{-3}]: electron density
ni [10^{20} m^{-3}]: ion density (=ne, at this moment)
Te [keV]: electron temperature
Ti [keV]: ion temperature
Pin_e [MW/m^3]: power input density to electron (currently only NBI)
Pin_i [MW/m^3]: power input density to ion (currently only NBI)
heat_flux_e [MW/m^2]: electron energy flux
heat_flux_i [MW/m^2]: ion energy flux
heat_flux_in_e [MW/m^2]: electron energy flux (corresponding only to heating source such as NBI)
heat_flux_in_i [MW/m^2]: ion energy flux (corresponding only to heating source such as NBI)
inte_Pe [MW]: volume-integrated Pe
inte_Pi [MW]: volume-integrated Pi
inte_Pin_e [MW]: volume-integrated Pin,e
inte_Pin_i [MW]: volume-integrated Pin,i
inte_Pie [MW]: volume-integrated Pie
inte_Pei [MW]: volume-integrated Pei (=-inte_Pie)
dTe/drho: electron temperature gradient

dTi/drho: ion temperature gradient
dne/drho: electron density gradient
dni/drho: ion density gradient
<lnabla rhol>: $\langle|\nabla\rho|\rangle$
<lnabla rhol^2>: $\langle|\nabla\rho|^2\rangle$
dV/drho: $dV/d\rho(=V')$
inte_Pin_e_nbi1 [MW]: volume-integrated deposited power to electron and ion from NBI#1-5
inte_Pin_i_nbi1 [MW]
inte_Pin_e_nbi2 [MW]
inte_Pin_i_nbi2 [MW]
inte_Pin_e_nbi3 [MW]
inte_Pin_i_nbi3 [MW]
inte_Pin_e_nbi4 [MW]
inte_Pin_i_nbi4 [MW]
inte_Pin_e_nbi5 [MW]
inte_Pin_i_nbi5 [MW]

3.6 Dytrans [4]

H.Lee, K.Ida, M.Yoshinuma

dytrans evaluates “dynamic transport”, in which energy flows due to the temporal variation of plasma profiles are also taken into account.

3.6.1 Input

eg:fit3d_sd, cxsmap6(7), tsmap, [***ermap6(7)***, if exists]

3.6.2 Output

eg: dytrans6_sd or ***dytrans7_sd*** (using cxs6 or cxs7)
(due to the faster temporal resolution in cxs6 that cxs7, dytrns6_sd is appropriate)

DimNo = 2
Time [s]
R [m]
ValNo = 88

reff [m]: minor radius
rho: reff/a99

Ti [keV]: ion temperature
Ti_fit [keV]: ion temperature profile fitted with polynomial function
Tier [arb]: error of ion temperature
Te [keV]: electron temperature
Te_fit [keV]: electron temperature profile fitted with polynomial function
Vc [km/s]: toroidal rotation velocity
Vcer [arb] error of toroidal rotation velocity
<Er> [kV/m]: flux-averaged radial electric field
Ne [10^{19} m^{-3}]: electron density
Ne_fit [10^{19} m^{-3}]: electron density profile fitted with polynomial function

<u>Gradient /dR</u>
dTi/dr [keV/m]: dTi/dR
dTi/dr_fit [keV/m]: dTi_fit/dR
dTe/dr [keV/m]: dTe/dR
dTe/dr_fit [keV/m]: dTe_fit/dR
dVc/dr [10^3/s]: dVc/dR
d<Er>/dr [kV/m^2]: d<Er>/dR

dNe/dr [10^{19} m^{-4}]:	dNe/dR
dNe/dr_fit [10^{19} m^{-4}]:	dNe_fit/dR

<u>Gradient /dreff</u>

dTi/dreff' [keV/m]:	dTi/d(reff)
dTi/dreff_err [keV/m]	
dTi/dreff_fit [keV/m]	
dTe/dreff [keV/m]:	dTe/d(reff)
dTe/dreff_fit [keV/m]	
dVc/dreff [10^3/s]:	dVc/d(reff)
dVc/dreff_err [10^3/s]	
d<Er>/dreff [kV/m^2]:	d〈Er〉/d(reff)
dNe/dreff [10^{19} m^{-4}]:	d(ne)/d(reff)
dNe/dreff_fit [10^{19} m^{-4}]	

<u>Gradient /drho (rho=reff/a99)</u>

dTi/drho [keV]:	dTi/d(rho)
dTi/drho_fit [keV]	
dTe/drho [keV]:	dTe/d(rho)
dTe/drho_fit [keV]	
dVc/drho [km/s]:	dVc/d(rho)
d<Er>/drho [kV/m]:	d〈Er〉/drho
dNe/drho [10^{19} m^{-3}]:	dne/d(rho)
dNe/drho_fit [10^{19} m^{-3}]	

<u>Time derivative terms</u>

dTi/dt [keV/s]:	dTi/dt
dTi/dt_fit [keV/s]:	
dTe/dt [keV/s]:	dTe/dt
dTe/dt_fit [keV/s]	
dVc/dt [km/s^2]:	dVc/dt
d<Er>/dt [kV/(ms)]:	d〈Er〉/dt
dNe/dt [10^{19} m^{-3}/s]:	dne/dt
dNe/dt_fit [10^{19} m^{-3}/s]	

Wpi [kJ]:	volume averaged plasma ion energy with Ti and Ne
Wpi_fit [kJ]:	volume averaged plasma ion energy with Ti_fit and Ne_fit
Wpe [kJ]:	volume averaged plasma electron energy with Te and Ne
Wpe_fit [kJ]	volume averaged plasma electron energy with Te_fit and Ne_fit

Wpiflx [kJ/m^2]:	Wpi/S(reff)
Wpiflx_fit [kJ/m^2]:	Wpi_fit/S(reff)
Wpeflx [kJ/m^2]:	Wpe/S(reff)
Wpeflx_fit [kJ/m^2]:	Wpe_fit/S(reff)

<u>Change in heat flux due to the change in temperature and density</u>

delQi/Ne [keV m/s]:	
delQi/Ne_fit [keV m/s]:	with Ti_fit and Ne_fit
delQe/Ne [keV m/s]:	
delQe/Ne_fit [keV m/s]:	with Ti_fit and Ne_fit

<u>Heat flux normalized density ion-electron energy exchange included</u>
(QinbioverNe and QexoverNe are not included in the output file)

Qi/Ne [keV m/s]:	QinbioverNe - QexoverNe
Qi/Ne_fit [keV m/s]:	QinbioverNe_fit-QexoverNe_fit
Qe/Ne [keV m/s]:	QinbioverNe + QexoverNe
Qe/Ne_fit [keV m/s]:	QinbioverNe_fit+QexoverNe_fit

<u>Thermal diffusivity (steady state)</u>

Chi_i [m^2/s]:	(Qi/Ne+delQi/Ne)/(dTi/dreff)
Chi_i_fit [m^2/s]:	(Qi/Ne_fit+delQi/Ne_fit)/(dTi/dreff_fit)

Chi_e [m^2/s]: (Qe/Ne+delQe/Ne)/(dTe/dreff)
Chi_e_fit [m^2/s]: (Qe/Ne_fit+delQi/Ne_fit)/(dTe/dreff_fit)

<u>Heat flux normalized density ion-electron energy exchange included and delta Q/n</u>
Qi/Ne+delQi/Ne [keV m/s]:
Qi/Ne_fit+delQi/Ne_fit [keV m/s]:
Qe/Ne+delQe/Ne [keV m/s]:
Qe/Ne_fit+delQe/Ne_fit [keV m/s]:

<u>Thermal diffusivity include delta_Q</u>
Chi_i_dy [m^2/s]: -(Qi/Ne+delQi/Ne)/(dTi/dreff)
Chi_i_dy_fit [m^2/s]: -(Qi/Ne_fit+delQi/Ne_fit)/(dTi/dreff_fit)
Chi_e_dy [m^2/s]: -(Qe/Ne+delQe/Ne)/(dTe/dreff)
Chi_e_dy_fit [m^2/s]: -(Qe/Ne_fit+delQe/Ne_fit)/(dTe/dreff_fit)

<u>Torque density</u>
P/MiNe [e3 m^2/s^2]
P/MiNe_fit [e3 m^2/s^2]

<u>Perpendicular viscosity</u>
mu_tor [m^2/s^2]
mu_tor_fit [m^2/s^2]

<u>Ion heat flux only due to NBI normalized by fitted electron density</u>
QinbioverNe_fit [keV m/s]

<u>Electron heat flux only due to NBI normalized by fitted electron density</u>
QenbioverNe_fit [keV m/s]

<u>Ne_fit normalized ion-electron energy exchange calculated with Ti_fit, Te_fit, and Ne_fit</u>
QexoverNe_fit [keV m/s]

<u>Temperature ratio</u>
Te/Ti
Te_fit/Ti_fit

<u>Normalized thermal diffusivity</u>
Chi_i_dy/Ti^3/2 [$m^2/s/keV^{1.5}$]
Chi_i_dy/Ti^3/2_fit [$m^2/s/keV^{1.5]}$
Chi_e_dy/Te^3/2 [$m^2/s/keV^{1.5}$]
Chi_e_dy/Te^3/2_fit [$m^2/s/keV^{1.5}$]

<u>Surface</u>
dV/dreff' [m2]: dV/d(reff) = S(reff) : consistency checked with that in TRsnap

_fit : using fitting curve (functional form is same as that for fit3d)

References

[1] C.Suzuki et al., to appear in Plasma Phys. Control. Fusion (2012).
[2] S. Murakami et al., Trans. Fusion Technol. **27** (1995) 256.
[3] R.Seki et al., Plasma Fusion Res. **6** (2011) 2402081
[4] H.Lee, K.Ida et al., to appear in Plasma Phys. Control. Fusion (2012).

[Appendix] shell and input for fit3d & trsnap

input.trsnap_XXXXXXtYYYY.txt (T:\trsnap\XXXXXX\XXXXXXtYYYY)

```
&modules
mod_name[1] = 'fit3d_2010_1'
mod_name[2] = 'tr_snap_2010'
&end
```

```
&comp_env
local_work_dir = './work_dir/'
local_code_dir = './bin/'
comp_name[1] = 'task4lhd'
comp_ip[1] = 'task4lhd02.LHD.nifs.ac.jp'
comp_user[1] = 'yokoyama'
comp_work_dir[1] = '/work2/yokoyama/W_W'
comp_code_dir[1] = '/work/seki/TASK4LHD'

local_result_dir = 'T:\trsnap\109027\109027t5800'
plus_name='109027t5800'
&end

&graphic
graphic = 'on'
graphic_file = 'result.ps'
graphic_close='on'
&end

&finalize_process
delete_local_tmp_dir = 'no'
delete_remote_work_dir = 'no'
&end

&t3dparam
B0= 1.38,
rmaj0= 3.75,
ra =  0.59403,
zi=1
&end

&neprof
nepfn='T:\trsnap\109027\109027t5800\ne_109027t5800.txt',          Point-series based on fitting
;;nepfun='(ne0-nea)*(1.0-rho**2)+nea', ne0=1.0e1, nea=0.0e-2, ;; (10e19m-3)     Functional form
&end

&teprof
tepfn='T:\trsnap\109027\109027t5800\Te_109027t5800.txt',
;;tepfun='(te0-tea)*(1.0-rho**2)+tea', te0=3.0e0, tea=0.0e-2,;; (keV)
&end

&zeffprof
zeff=1, zimp=6,
zeffpfn='',
zeff_pfun='', zeff0=2.0, zeffa=2.0,                 zeff=2 (radially constant) is assumed by default
&end

&tiprof
tipfn='T:\trsnap\109027\109027t5800\Ti_109027t5800.txt',
;;tipfun='', ti0=1.0e0, tia=1.0e-2,;; (keV)
&end

&timpprof
timppfn='',
timppfun='', timp0=1.0e0, timpa=1.0e-2, ;; (keV)
&end

&fit3dparam
ipg=0
npart=103000
cn0=0.0
newboz_comp='local'
newboz_file='T:\equilibrium\109027\109027t5800\newboz_b_109027t5800.dat'   newboz data
inbi1='on'
enbi1=  0.0000, pnbi1=  1.0000
inbi2='on'
enbi2=  0.0000, pnbi2=  1.0000
```

```
inbi3='on'
enbi3=176.4790, pnbi3=  1.0000
inbi4='on'
enbi4=  0.0000, pnbi4=  1.0000
inbi5='on'
enbi5=  0.0000, pnbi5=  1.0000
&end

&fit3d_output
fit_data.out = 'no'
fit_data.out10 = 'yes'
fit_data.out11 = 'no'
fit_data.out20 = 'yes'
fit_data.out21 = 'no'
fit_data.out30 = 'yes'
fit_data.out31 = 'no'
fit_data.out40 = 'yes'
fit_data.out41 = 'no'
mcnbi_data.out = 'no'
data.out6 = 'yes'
data.out7 = 'yes'
data.out9 = 'yes'
data.out15 = 'no'
data.out20 = 'no'
data.out21 = 'yes'
data.out30 = 'no'
data.out40 = 'no'
&end

&tr_snap_param
nrmax=60
wout_comp='local'
wout_file='T:\equilibrium\109027\109027t5800\wout_109027t5800.vmec'          wout
trparam_file=''
;;nbi_power_file[1]=''
;;nbi_power_file[2]=''
;;nbi_power_file[3]=''
;;nbi_power_file[4]=''
;;nbi_power_file[5]=''
;;ech_power_file[1]=''
;;ich_power_file[1]=''
;;rloss_file[1]=''
&end

&tr_snap_output
tr_snap.101='yes'
tr_snap.102='yes'
tr_snap.111='yes'
tr_snap.112='yes'
tr_snap.113='yes'
&end
```