TGLFEP code document

How to run TGLF with energetic particles

He Sheng [1]
Peking University

Directed by R. E. Waltz
General Atomics

May 11, 2017

[1] Email: hesheng.phy@pku.edu.cn

Contents

1 Critical EP beta by TGLFEP code 3

2 Introductory to TGLFEP code 5

2.1 Input: profiles 5

2.2 Input: control parameters 8

2.3 MODE_IN 9

2.4 PROCESS_IN 10

2.5 FACTOR_IN 11

2.6 WIDTH_IN 11

2.7 KY_MODEL 11

2.8 Parametric scan 12

Start case

We started the simulations with GA-std ITER-α case, which has the parameters:

- $r/a = 0.5, R/a = 3, q = 2, \hat{s} = 1, \alpha = 0,$
- $a/L_{T_e} = a/L_{T_i} = 3, a/L_{n_e} = a/L_{n_i} = 1, T_i/T_e = 1,$
- $a/L_{n_{EP}} = 4, a/L_{T_{EP}} = 0, n_i + 2n_{EP} = n_e, n_{EP}/n_e = 2.5\%,$
- $\beta_e = 0.002, T_{EP}/T_e = 100.$

To get the AE growth rate (and frequency) versus $k_y = k_\theta \rho_s$ spectrum, we can directly run TGLF with the input file ('iput.tglf') as

```
GEOMETRY_FLAG = 0
NS            = 3
ZS_3          = 2.0
MASS_3        = 2.0
RLNS_3        = 4.0
RLTS_3        = 0.0
TAUS_3        = 100.0
AS_2          = 0.95
AS_3          = 0.025
BETAE         = 0.002
USE_BPER      = TRUE
FILTER        = 0.0
WIDTH         = 1.87
FIND_WIDTH    = FALSE
KYGRID_MODEL  = 0
KY            = 0.15
NKY           = 30
NMODES        = 4
NBASIS_MIN    = 32
NBASIS_MAX    = 32
NXGRID        = 32
```

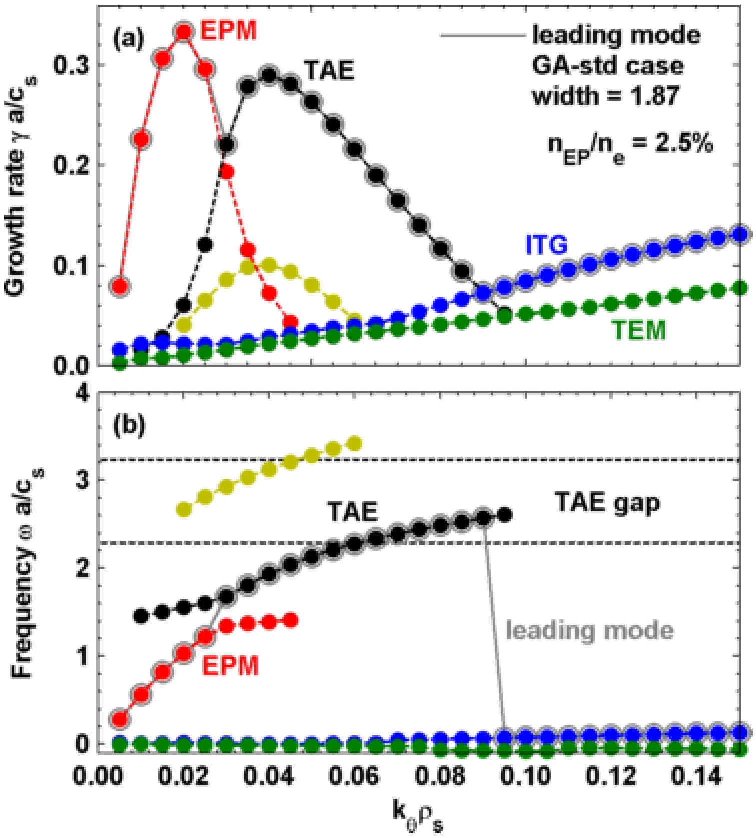

Figure 1: Linear eigenmodes spectrum for GA-std ITER-α case.

The detail meaning of each input parameters can be found in TGLF introduction webpage [https://fusion.gat.com/theory/Tglfinput].

More explanations on the NOT default inputs are given as below:

FILTER is set as 0.0 instead of 2.0 to simulate the high-frequency AE modes. If only ITG/TEM modes simulated, FILTER can set back to 2.0 [1];

WIDTH is an important TGLF parameter to simulate TAE/EPM modes. A constant WIDTH for the whole $k_\theta \rho_s$ spectrum, like 1.87 for GA-std ITER-α case, is found much successful with the comparison to GYRO AE simulations. So FIND_WIDTH should be set as FALSE;

NBASIS the number of basis functions should be set up to 32, so NBASIS_MIN, NBASIS_MAX, NXGRID are 32.

[1] Tips: to plot the whole ITG/TEM spectrum in Figure. 1, re-run with 'FILTER = 2.0' again.

1

Critical EP beta by TGLFEP code

This chapter is about how to get the critical EP beta β_{EP}^{crit} for the DIII-D NBI case using a new TGLFEP code. For the details of TGLFEP code, see Chapter 2.

TGLFEP code is a program to run TGLF with EP linear simulations conveniently, where recalling TGLF (inside GAcode) with the version number (on May 7th, 2016) "TGLF stable_r6.0.0-31-gba49" on NERSC "EDISON_CRAY Linux x86_64" or "CORI Linux x86_64"

The latest version of TGLFEP has been updated to GitHub [1]. You can download it directly or use the Git command:

```
$ git clone git@github.com:heshengpku/TGLFEP.git
```

to clone it to your own workspace. Then just use the command 'make' for makefile.

There are two input files for TGLFEP code, 'input.profile' (details discussed in Section 2.1) and 'input.TGLFEP' (details discussed in Section 2.2). The 'input.profile' file contains the plasma and geometry (radial) profiles in consistent with the TGLF formats. The 'input.TGLFEP' file contains the control parameters of TGLFEP.

These two files (example case) has exactly existed after you download TGLF source files. The example profiles are for the DIII-D NBI case with 50 radii (r/a = 0.0 excluded), which are calculated by EPtran code . The 'input.TGLFEP' has been set to get the critical EP beta (profile) for the inner 40 radii. The outputs are in the file 'out.TGLFEP'. For the outside 10 radii, save the above output files at first and just change '40 SCAN_N' to '10 SCAN_N', '1 IRS' to '41 IRS', and '4.0 FACTOR_IN' to '8.0 FACTOR_IN' in 'input.TGLFEP' file. Then run TGLFEP again. The final result will look like Figure 1.1.

[1] https://github.com/heshengpku/TGLFEP

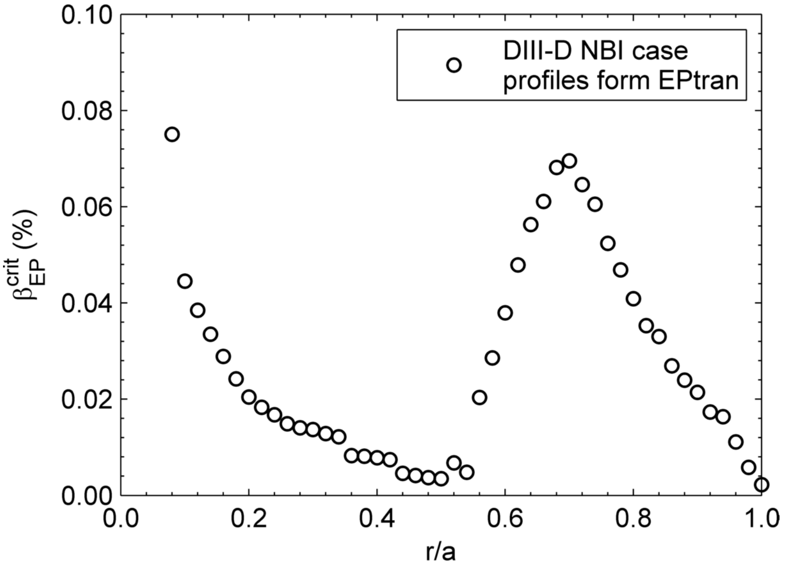

Figure 1.1: The EP critical beta for the DIII-D NBI case where the profiles are from EPtran code.

'SCAN_N' is the amount of radii to calculate (don't have to be the same with the total number of radii, but should be less than or equal to that). 'IRS' is the starting radius.

'FACTOR_IN' indicates the upper boundary of the EP density scale. If the input EP density cannot drive AE modes, a larger FACTOR_IN > 1.0 (up to 2., 10. or even 100.) should be used.

When you want to run a new case with TGLFEP, for example still the DIII-D NBI case but with different profiles.

- At first, you should generate a new 'input.profile' based on your plasma and geometry profiles in the TGLF formats.
- Then try to run TGLFEP code and get the new output file.
- Check the outputs. If at some radii, EP critical beta are not found (that is, SFmin shows NOT correctly in 'out.density_threshold'), you need to change 'FACTOR_IN' to a larger value and try for these radii again.

By the way, however it is possible that for some cases, the EP critical beta doesn't exist and surely cannot be found even if with a very large 'FACTOR_IN'.

Recommend to use 25×SCAN_N CPUs (or more) to run these cases. It usually costs about 20 minutes on Cori (using ceil(25×SCAN_N/32) nodes) or on Edison (using ceil(25×SCAN_N/24) nodes).

2

Introductory to TGLFEP code

TGLFEP source files are listed as below

```
TGLFEP_interface.f90
TGLFEP_tglf_map.f90
TGLFEP_ky.f90
TGLFEP_TM.f90
TGLFEP_ky_widthscan.f90
TGLFEP_ky_nEPscan.f90
TGLFEP_scalefactor.f90
TGLFEP_mainsub.f90
TGLFEP_driver.f90
```

'TGLFEP_interface.f90' contains the module of internal parameters for TGLFEP code.

'TGLFEP_tglf_map.f90' contains the module of input profiles and the subroutine to read plasma and geometry profiles from input file 'input.profile' as well as the subroutine to map the parameters to TGLF in the required formats.

'TGLFEP_driver.f90' will read the control parameters from input file 'input.TGLFEP'. It will also write down the output file.

The other subroutines contain different processes for different simulation schemes as discussed in Section 2.4.

2.1 Input: profiles

There are four different example profiles that have been studied by TGLFEP code as listed in Table 2.1. Their input files are currently saved in the folder 'input'.

Table 2.1: Input profiles

Input cases tested	Descriptions
GA-std ITER-α case	The start case
DIII-D NBI case	DIII-D profiles as the same as GYRO inputs
EPtran profiles	EPtran code output profiles without transport
Two EP species case	3.5MeV alpha particle and 1MeV NBI in GA-std case

The whole data of the input profiles are read from the file 'input.profile'. The following is the example for the GA-std ITER-α case.

```
1.0  SIGN_BT
1.0  SIGN_IT
1  NR
3  NS
0  GEOMETRY_FLAG
---------------------------------------------------------------
# electron species
-1.0  ZS
2.723D-4  MASS m/m_D
# normalized density gradients: rlns
1.0
# normalized temperature gradients: rlts
3.0
---------------------------------------------------------------
# ion species 1
1.0  ZS
1.0  MASS m/m_D
# normalized density: as
0.95
# normalized temperature: taus
1.0
# normalized density gradients: rlns
1.0
# normalized temperature gradients: rlts
3.0
---------------------------------------------------------------
# ion species 2
2.0  ZS
2.0  MASS m/m_D
# normalized density: as
0.025
# normalized temperature: taus
```

```
100.0
# normalized density gradients: rlns
4.0
# normalized temperature gradients: rlts
0.0
---------------------------------------------------------------
# Geometry
# minor radius: rmin
0.5
# major radius: rmaj
3.0
# safety factor: q
2.0
# magnetic shear: shear
1.0
# normalized pressure gradient: alpha
0.0
---------------------------------------------------------------
# effective ion charge: zeff
1.0
# betae
0.002
# ky = (nq/r)*rho_s for n = 1
0.005
```

NR is the number of radii and NS is the number of species. All the input parameters should be in the TGLF formats, i.e. gyro-Bohn normalization.

For the input order of species, the first species should be always electron and its normalized density and temperature have been set as default 1. The second species (i.e. ion species 1) should be always the main ions, e.g. it is deuterium for GA-std ITER-α case.

From the third species (ion species 2 and/or more) are the energetic particles (EP species). The last species is set as variable EP density to find the density threshold or to do other things. The scaled EP density is multiplied by a parameter FACTOR_IN.

Besides the necessary TGLF inputs, the parameters ‘k_y for n = 1’ [1] is also needed by TGLFEP code. k_y is used to calculate different toroidal number n. However, it is actually useless for GA-std case, more discuss in Section 2.7. By the way, the low frequency modes are filter out by $|\omega| < 0.2 \times \omega_{TAE}$ when looking for AE growth rate $\gamma_{AE} = 0$.

For the geometry, only $s - \alpha$ geometry (GEOMETRY_FLAG = 0) and Miller geometry (GEOMETRY_FLAG = 1) have been used in TGLFEP code. If Miller geometry is chosen, the geometry inputs need more profiles

[1] $k_y = k_\theta \rho_s = (nq/r) \times \rho_s$

as in sequence: ‘rmin’, ‘rmaj’, ‘q’, ‘shear’, ‘q_prime’, ‘p_prime’ [2], ‘shift’, ‘kappa’, ‘s_kappa’, ‘delta’, ‘s_delta’, ‘zeta’, ‘s_zeta’.

Other TGLF input paramters of plasma and geometry profiles which are not included in TGLFEP ‘input.profile’ will use the TGLF defaults.

An important thing to note here is when to write a new ‘input.profile’ file, the parameters should keep in order, and the comment lines in ‘input.profile’ should be kept just like the example shown to avoid the input errors.

To check if the input profiles are read correctly or not, you can remove the comment of the line

```
! if(id .eq. 0) call dump_profile
```

in ‘TGLF_driver.f90’ to print a dump file ‘dump.profile’. If correctly, the data of dump file ‘dump.profile’ should be exactly the same with the input file ‘input.profile’.

2.2 Input: control parameters

The control parameters of TGLFEP code are read from file ‘input.TGLFEP’. There are usally two operational modes recommended:

- AE growth rate spectrum: an example ‘input.TGLFEP’ to get growth rate and frequency spectra for GA-std ITER-α case

  ```
  3  PROCESS_IN
  0  KY_MODEL

  1  SCAN_N
  1  IRS

  .false. FACTOR_IN_PROFILE
  1.0  FACTOR_IN
  .true.  WIDTH_IN_FLAG
  1.87  WIDTH_IN
  ```

  The result is exactly Figure 1. The outputs are in the files ‘out.eigenvalue_m*’, where ‘m*’ means different MODE_IN as discussed in Section 2.3.

- Density threshold (critical EP beta) for $\gamma_{AE} = 0$: an example ‘input.TGLFEP’ file to get the density threshold n^{th}_{EP}/n_e versus safety factor q for GA-std ITER-α case

[2] Note: as the definition in TGLF, p_prime should be usually negative. Also directly confirmed by Gary Staebler.

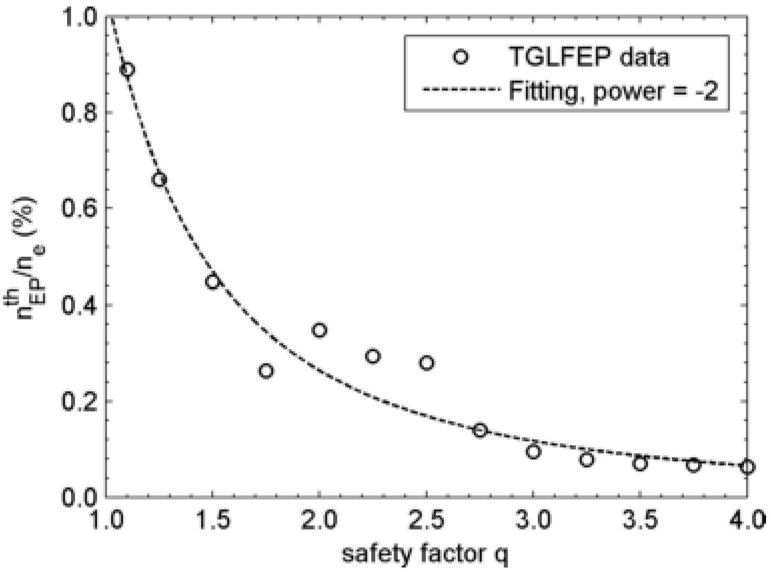

Figure 2.1: Density threshold versus safety factor q for GA-std ITER-α case.

```
4  PROCESS_IN
0  THRESHOLD_FLAG
0  KY_MODEL

13  SCAN_N
1  IRS

.false.  FACTOR_IN_PROFILE
1.0  FACTOR_IN
.false.  WIDTH_IN_FLAG
1.5  WIDTH_MIN
2.0  WIDTH_MAX
```

The result will look like Figure 2.1. The outputs are in the file 'out.TGLFEP', where 'SFmin' means the minimum scale factor. The EP density threshold n_{EP}^{th}/n_e, and the EP critical beta β_{EP}^{crit} are also given.

The meaning and the input format of each parameters are discuss in the following sections. Note that the input control parameters should be kept in order correctly in 'input.TGLFEP' file.

2.3 MODE_IN

We usually discussed different growth rates for critical EP gradient model as MODE_IN =

1: $\gamma_{AE+ITE/TEM}$, the AE growth rate with the total plasma background gradients.

2: γ_{AE}, the AE growth rate without the background gradients, i.e. only EP gradients work.

3: $\gamma_{ITG/TEM}$, the ITE/TEM growth rate without the EP gradients.

4: is the same as 1 but setting FILTER = 2.0. We can get the ITE/TEM growth rate with the EP gradients and the background gradients. It is actually the method used in Figure 1 to get the whole ITG/TEM spectrum.

2.4 PROCESS_IN

The input parameter PROCESS_IN is used in 'mainsub' subroutine to clarify several different simulation scheme as PROCESS_IN =

0: 'TGLFEP_ky' to get the modes' wavefunction at a given k_y [3];

1: 'TGLFEP_TM' to get growth rate and frequency spectra with a k_y spectrum [4];

2: 'TGLFEP_ky_nEPscan' to get growth rate and frequency versus n_{EP} at a single marked k_y;

3: (Recommend) to get growth rate and frequency spectra (and/or the growth rate versus n_{EP}). It can run with a given WIDTH or find a suitable WIDTH in a given range at a marked k_y;

4: (Recommend) to find the EP density threshold of $\gamma_{AE} = 0$

[THRESHOLD_FLAG = 0] for multiple toroidal n (equivalently k_y) to get a minimum density threshold and the EP critical beta (%),

[THRESHOLD_FLAG = 1] for a series $a/L_{n_{EP}}$ which can be used to calculate C_R,

with a given WIDTH or find a suitable WIDTH in a given range at a marked k_y.

For PROCESS_IN = 0 or 1, MODE_IN should be specified at the second line in 'input.TGLFEP' file just as Section 2.3 clarified.

For PROCESS_IN = 4, THRESHOLD_FLAG should be set at the second line in 'input.TGLFEP' file.

[3] A input file 'input.ky' needed which contains the value of the given k_y

[4] Default $0 < k_y \leq 0.15$ and $\Delta k_y = 0.01$. The k_y spectrum can also be changed by revising the module 'ky_spectrum' in 'TGLFEP_TM.f90' file

2.5 FACTOR_IN

For PROCESS_IN NOT = 4, FACTOR_IN is multiplied to the EP density read from the input profile (so FACTOR_IN = 1.0 will use the input value of EP density).

For PROCESS_IN = 4, FACTOR_IN indicates the upper boundary of the EP density scale. If the input EP density cannot drive AE modes, a larger FACTOR_IN > 1.0 (up to 2., 10. or even 100.) should be used.

'FACTOR_IN_PROFILE' is read before 'FACTOR_IN', where FACTOR_IN_PROFILE = TRUE means FACTOR_IN will read-in as a profile, while FACTOR_IN_PROFILE = FALSE means only one FACTOR_IN will read-in and it will be the same for each radius.

2.6 WIDTH_IN

WIDTH_IN_FLAG is 'TRUE' for using a constant given WIDTH, or 'FALSE' for finding the WIDTH in a range [WIDTH_MIN, WIDTH_MAX) to maximize γ_{AE} at a marked k_y. About the marked k_y, detail in Section 2.7.

For WIDTH_IN_FLAG = TRUE, the input WIDTH_IN should be given explicitly at the end of 'input.TGLFEP' file and it should be a profile with SCAN_N points.

For WIDTH_IN_FLAG = FALSE, WIDTH_MIN and WIDTH_MAX should be both given in 'input.TGLFEP'. Then TGLFEP code will run WIDTH scan in step = 0.01 [5] in the range [WIDTH_MIN, WIDTH_MAX) to find the maximum γ_{AE} [6]. The final WIDTH chosen by TGLFEP code is printed out in file 'out.TGLFEP'.

Note that WIDTH_IN_FLAG should be always 'TRUE' and WIDTH_IN should be given in 'input.TGLFEP' file for PROCESS_IN = 0, 1 or 2.

2.7 KY_MODEL

KY_MODEL determines the relationship between the integer n and k_y as KY_MODEL =

0: $k_y = 0.01n$ usually only for GA-std case;

1: $k_y = nq/(r/a) \times \rho_*$, here n is exactly the toroidal number;

2: k_y is equal to where $k_\theta \rho_{EP} = 0.1n$, This is available to avoid $k_\theta \rho_{EP} > 1.0$ problem in KY_MODEL = 1.

[5] The step is chosen small because γ_{AE} versus WIDTH may be peak in some cases.

[6] Details in the source file 'TGLFEP_ky_widthscan.f90'.

The marked k_y is corresponding to $n = 4$ for KY_MODEL = 0 and $n = 3$ for KY_MODEL = 1 or 2, when looking for WIDTH maximizing γ_{AE} (WIDTH_IN_FLAG = FALSE).

So the input 'k_y for n = 1' in 'input.profile' only works for KY_MODEL = 1.

2.8 Parametric scan

TGLFEP can use MPI to do the parametric scan. The number of parametric scan is set as SCAN_N.

For radial scan, the starting radius (IRS) is read-in just after the line of SCAN_N [7]. For other parametric scan, like safety factor q scan and shear $\hat{s}$ scan, just generate a SCAN_N profile in 'input.profile'.

[7] For an example, if SCAN_N = 3 and IRS = 2, the simulation will run at the 2, 3, 4 radial grids of the input profiles