GENE and its pre-processing and post-processing

Max T. Curie

May 21, 2021

Abstract

The goal of this exercise is to introduce GENE and its pre and post processing tool that have been developed in the plasma community.

Contents

Chapter 1

Introduction and General Background Knowledge

1.1 Basic of terminal commands

1.1.1 Public key

Use the following command to check the public key

```
ls -al ~/.ssh
```

Here is the link on how to generate a public key.

```
https://confluence.atlassian.com/bitbucketserver059/creating-ssh-keys-94925482
```

If one has a existing public key, use the following command.

```
vi ~/.ssh/id_rsa.pub
```

1.1.2 Command

Here are some common command line on the terminal
To cancel the job.

```
nersc$ scancel $jobid
```

Permission clearance of the directory:

```
chmod a+rx<global/csratch1/sd/maxcurie>
```

Unzip

```
tar xvf filename.tar
```

show available modules

```
module avail python
```

1.1.3 GitHub

To download from GitHub

```
git clone https://github.com/drdrhatch/IFS_scripts.git
```

To update the folder

```
git pull
```

If the update does not work, one can try reset it.

```
git reset
```

One can commit the

```
git add example.py
git commit -m "Comment"
git push origin master
```

One may need to add public key, for more information, please check the link below.

```
https://help.github.com/en/github/authenticating-to-github/connecting-to-githu
```

1.1.4 Vi

One need to access to the output directory and

```
grep kymin n0_12/parameters_1
```

1.1.5 Protect your file from being deleted

Check the following link:

```
https://docs.nersc.gov/filesystems/archive/
```

1.1.6 Shortcuts

By entering the following line, one can establish custom commands

```
vi ~/.bashrc.ext
```

The edit or add the alias to the file and save it. Remember DO NOT leave space after "=" sign or it won compile. Here are some useful shortcuts.

```
alias edit="vi ~/.bashrc.ext"
alias cds="cd $SCRATCH"
alias source="source ~/.bashrc.ext"
alias msubmit2="sbatch submit_knl.cmd"
alias mdebug="sbatch submit_debug.cmd"
alias mtime="squeue -u maxcurie"
alias mkill="scancel -u maxcurie"
alias start0="module load python"
alias plot_E_t="$HOME/genecode/tools/gplot"
alias global0="$HOME/genecode/tools/runassign 1"
alias meigen="python $HOME/max/scripts/xing_TestDrive_fieldsWrapper3.py
            -p T -a T"
```

1.2 The toolbox

The script repository can be found here on GitHub:

```
https://github.com/drdrhatch/IFS_scripts
```

This repository contains scripts that are used to generate input profiles that are compatible with GENE formatting and process/visualize GENE output.

1.2.1 Cross reference in differenet folders

To manage the scripts from different folders and cross reference Here is a good tool:

```
# some_file.py
import sys
# insert at 1, 0 is the script path (or '' in REPL)
sys.path.insert(1, '/path/to/application/app/folder')
```

1.2.2 Translate the python2 script to python3

Use the following line on terminal to translate the script from python2 to python3

```
2to3 -w example.py
```

If one needs to convert all the the script, use the following line

```
2to3 -w *.py
```

1.3 Common run time of the simulation

1.4 Common trouble shooting

1.4.1 No such file or directory

Python will only regarding the Python software directory as the directory to refer to. In order to use some of the script for GENE processing. For instance "sacn info efit.py" will give you an error "OSError:[Errno 2] No such file or directory"

One need to set the path of the script to one of the default path to be called for. One can achieve it by adding the following line to bash fill (use "vi /.bashrc.ext" to edit the file)

```
PATH=$PATH:/global/homes/letter/username/scripts
```

Then source the bash file by typing

```
source ~/.bashrc.ext
```

1.5 NoMechine

In order to have a more efficient working environment, Having NoMachine/NX installed is highly recommended. Here is the link to on how to install NoMachine/NX:

`https://docs.nersc.gov/connect/nx/`

Remember to do step 9.
Here is the modified steps that works for me.

1. Choose "SSH" for protocol and click Continue.
2. Type in "nxcloud01.nersc.gov" for Host (leave the port set to 22) and click Continue.
3. Choose "Password"
4. You will need to edit one of the NoMachine config files on your local machine. YOU MUST EDIT THIS FILE WHILE NOMACHINE IS CLOSED/NOT RUNNING. First exit the NoMachine program and then edit

   `$HOME/.nx/config/player.cfg`

   and change the following key from library to native:

   `<option key="SSH client mode" value="native" />`

5. For login, the password will be password + MFA password

1.6 Flowchart of running simulation

1.6.1 Local simulation

The flowchart of running local simulation is shown as Figure 1.1

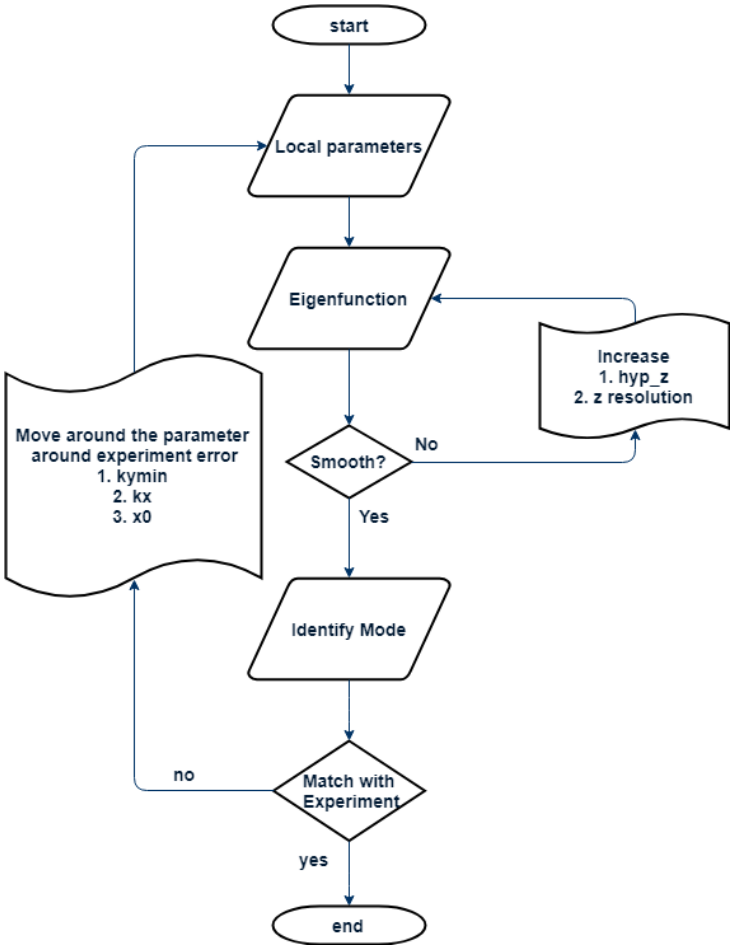

Figure 1.1: Flowchart of running local simulation

Low ky is hard to resolve, increase kymin will help the eigenfunction be to solved. The possible explanation is provided in the dispersion relation, section "Wave number".

By increasing hypz, running time will not increase. However, increasing z resolution will increase the computation time quadratically.

1.6.2 Global simulation

The flowchart of running local simulation is shown as Figure 1.2

By increasing hypz, u buffer size, ucoef krock, running time will not increase. However, increasing x, z resolution will increase the computation time quadratically.

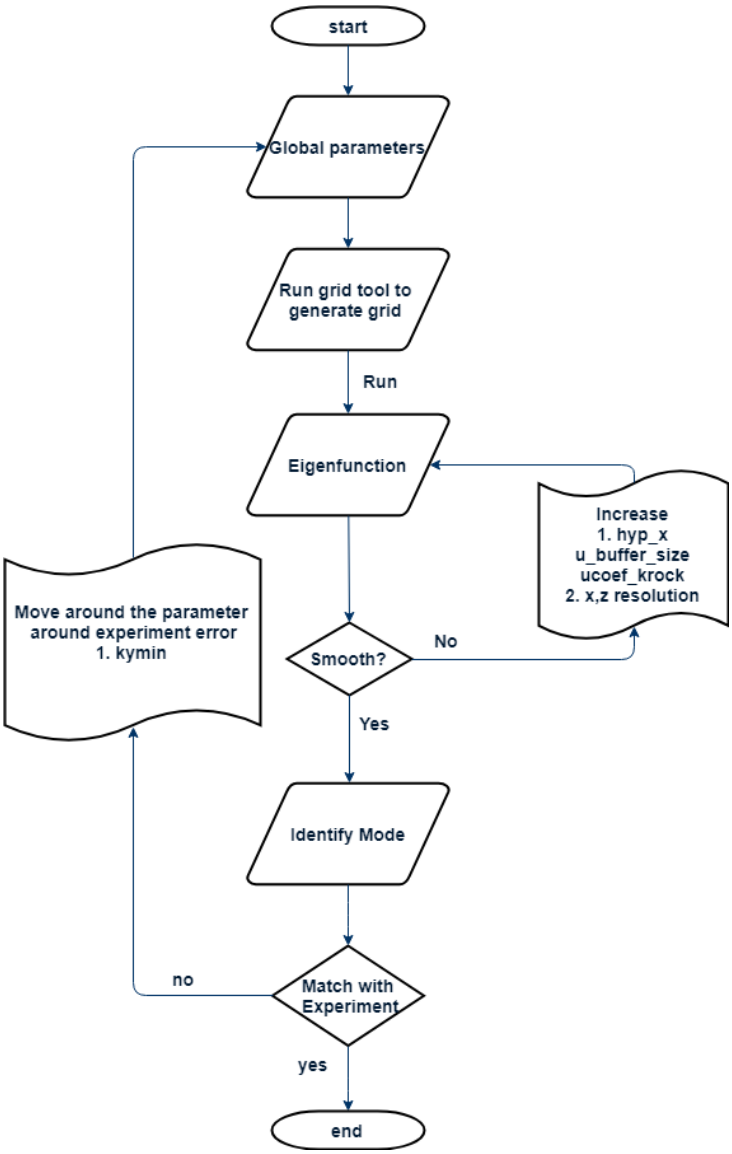

Figure 1.2: Flowchart of running global simulation

Chapter 2

How to run GENE

GENE's output is in Lab frame

2.1 System of euqations that GENE solves

$$\frac{\partial g}{\partial t} = Z + \mathcal{L}[g] + \mathcal{N}[g] \tag{2.1}$$

Where Z is a constant term that represent that curvature and density or temperature gradients. And $\mathcal{L}[g]$ is consisting of the drive term, the pressure term, a term describing the parallel dynamics, the curvature terms, the trapping term and collisions. And

$$Z = \frac{T_{0j}\left(2v_{\parallel}^2 + \mu B_0\right)}{q_j B_0} K_x \left(\omega_n + \left(v_{\parallel}^2 + \mu B_0 - \frac{3}{2}\right)\omega_{Tj}\right)\delta_{k_x,0}\delta_{k_y,0}F_{0j} \tag{2.2}$$

2.2 Discretization

The multipliers of k_x, k_y will be the grid of the perpendicular k space. z, $v_{||}$, ν will be discretized by dividing into finite grid.

$$\begin{aligned} &k_x = mk_x^{\min}, \quad k_y = nk_y^{\min} \\ &\text{with} \\ &k_x^{\min} = 2\pi/L_x, \quad k_y^{\min} = 2\pi/L_y, \quad m, n \in Z \end{aligned} \tag{2.3}$$

2.3 Boundary Condition

Periodic boundary condition is imposed for the calculation,

$$F\left(x+L_x,y,z\right)=F(x,y,z),\quad F\left(x,y+L_y,z\right)=F(x,y,z) \tag{2.4}$$

in Fourier space, we have

2.4 Theory of kinetic equation

$$\frac{d\rho}{dt}=\left(\frac{\partial}{\partial t}+\sum_j\left\{\vec{q}_j,H\right\}\cdot\frac{\partial}{\partial\vec{q}_j}+\sum_j\left\{\vec{p}_j,H\right\}\cdot\frac{\partial}{\partial\vec{p}_j}\right)\rho=0,j=1,\ldots,N \tag{2.5}$$

Recall that

$$\begin{cases}\dot{q}=\frac{\partial H}{\partial p}=\{q,H\}\\ \dot{p}=-\frac{\partial H}{\partial q}=\{p,H\}\end{cases} \tag{2.6}$$

The Hamiltonian can be expressed as:

$$H=\sum_j\left[\frac{p_j^2}{2m_j}+q_j\phi\left(\vec{x}_j\right)\right]+\sum_{i<j}\frac{q_iq_j}{\vec{x}_i-\vec{x}_j|} \tag{2.7}$$

After the gyro average, we has a guiding centered kinetic equation

$$\begin{aligned}&\frac{\partial F_j}{\partial t}+\left(v_\parallel\vec{b}_0+\frac{B_0}{B_{0\parallel}^*}\left(\vec{v}_{E\times B}+\vec{v}_{\nabla B_0}+\vec{v}_c\right)\right)\\&\quad\cdot\left(\vec{\nabla}F_j+\frac{1}{m_jv_\parallel}\left(q_j\vec{E}_1-\mu\vec{\nabla}\left(B_0+\overline{B}_{1\parallel}\right)\right)\frac{\partial F_j}{\partial v_\parallel}\right)=\langle C_j(F)\rangle\end{aligned} \tag{2.8}$$

This website explains the Landau collision operator

`https://farside.ph.utexas.edu/teaching/plasma/Plasma/node38.html#ey3.110a`

2.5 Mechanism of GENE

For more information, the following paper would be a great resource:

F. Merz "Gyrokinetic Simulation of Multimode Plasma Turbulence" (2008)

Guiding center approach is employed for the sake of simplicity.

The simulation is using momentum space instead of the spacial space to run the simulation, after the calculation, the reverse transformation was performed so

From section 2.2 "otherwise the boundary condition can lead to unphysical effects like very high values of transport due to end-to-end radial streamers of the electrostatic potential."

From section3.2 "physical processes that create smaller and smaller structures, once the lower limit is reached the finite resolution leads to the erroneous creation of again larger structures due to the sampling theorem"

2.6 Compile GENE

One need to Request a account from the website : http://genecode.org/

After the approval from GENE developer, use the following line to remove and (re)configure GENE, enter account and password, then start to download. Keep in mind the block structure grid is in the unstable version.

```
rm -rf genecode/
git clone https://username@gitta.rzg.mpg.de/~GENE/guest/git.py/gene.git
-b release-1.8 genecode
git clone https://usernam@gitta.rzg.mpg.de/~GENE/guest/git.py/gene.git
-b unstable genecode
```

To compile,

```
gmake -j
```

Compile in cori NERSC

Do not worry if you see "ERROR:152: Module 'craype-haswell' is currently not loaded". It will compile normally.

More info form NERSC: https://www.nersc.gov/assets/Uploads/06-Compiling-Codes.pdf

go to edit /bin/Cori.mk file and switch FUTILS = no to yes, and recompile the code by typing "gmake -j"

Compile in TACC Stampede2

If one used stampede2 in TACC, use the following commands,

```
cd makefiles
mkdir stampede2 (Make directory for stampede2)
cp -R ~/genecode/makefiles/stampede/stampede.mk ~/genecode/makefiles/stampede2
mv stampede.mk stampede2.mk

nano submit.cmd  (vi for CORI)
nano machine.def
gmake -j (make compile the changes)
```

Test run

```
cd ..
ls
./newprob  (getting a new prob folder)
ls

cd prob01
nano submit.cmd
```

Add those lines

```
#SBATCH -N 1           #Total node to use for the task
ibrun ./gene_stampede2
#SBATCH -A ITERP        # Which job is for ...
```

Submit a job

```
sbatch submit.cmd
```

TO check the status

```
squeue -u username
```

2.6.1 Compile with ability to read CHEASE profile

If one want to use CHEASE, one need to go under the folder makefile/Mechine/mechine.mk and change the following line:

```
FUTILS = yes
```

For more information about futils, On section 2.4 on GENE manual.

```
external/futils-gene/docs/futils.pdf
```

One need to load the following module before compile,

```
module swap craype-haswell craype-mic-knl
module load cray-petsc-complex
module load cray-hdf5-parallel
module load cray-fftw
module load gnuplot
module load craype
module load python
```

2.7 Local simulation

2.8 Global simulation

2.8.1 Basic mechanism of Global run

Linear Global run is the single ky, Different radial location and following field line.

Nonlinear is multiple ky

In order to countinue the run from the last run.

Make sure there is information of "checkoint" is not empty.

```
cp s_checkpoint checkpoint
run_assign 0001
```

Paraemeter file read checkpoint to "Ture".

2.8.2 Setup Global simulation

In order to run GENE more efficiently, one can compile a better version of the grid dividing mechanism.

From the email of Dr. David R Hatch:

Remeber to switch the specie so that electron(or species of interest) as the first species.

Get under the "genecode" folder, go to "tools/bsg prepost/grids"

First make sure the terminal has the python3 loaded. Here are several helpful commends

```
module load python3
module list intel
module swap python python3
```

Open README under current folder, run use the code in the folder to compile. Here are some example commands.

```
f2py -c --fcompiler=intelem -m gauss_quadrature gauss_quadrature.F90
```

After it is compiled, go to the prob folder. Make sure change the "prof type" to -1.

and add the following line to the end of the parameter file

```
 &bsgrid
is_bsg = T
wx_style = 'tight'
bbfdscheme = '4'
is_nv0_fixed = T
is_nw0_fixed = T
nblks = 4
vp_std = 4.0
opt_mthd = 'integral'
is_nblocks_bsg_fixed = T
blk_mks_r = 0.4, 0.55632126349, 0.66900662926, 0.783360067671, 0.9
lv_mks = 3.92453943632, 3.32938711669, 2.72743473838, 2.14143971412, 1.566474946
lw_mks = 18.6113156042, 13.3945543291, 8.98893860297, 5.54129644023, 0.3067304695
/
```

And make sure delete the scan lines at the end of parameter file and submit file. Anything that is behind the box section will resulting in a error. e.g. "nx0=256 !*******" will result in error. Then run the following code under the condition that python3 is loaded:

```
python3 ../tools/bsg_prepost/bsgrid.py cons parameters
```

Then change the prof type back to -2, and then submit the file.
Here is the example of the parameter file

2.8.3 Global run trouble shooting

Toroidal orientation

There maybe change from positive to negative or vis versa, One need to delete the line that specify the orientation of for the parameter file that GENE can determine the orientating from the iterdb file.

Intended mode does not appear

One has try a specific k and it works on the local linear simulation, but the reslut has a large discrepancy with what one would expect. As shown in Figure 2.1

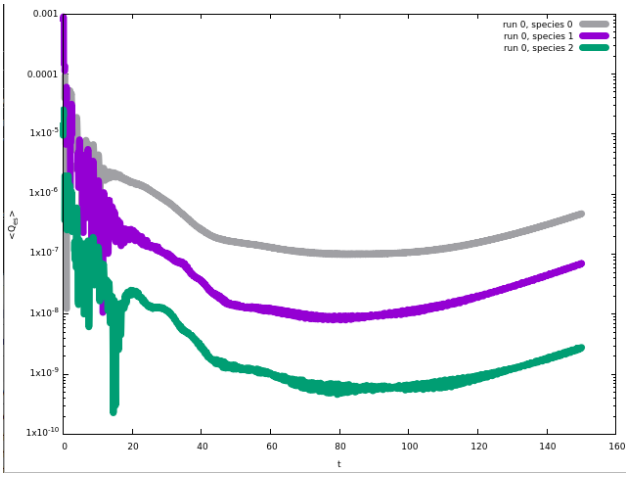

Figure 2.1: GENE global linear simulation give "no growth" on the expected mode.

One may need to run this simulation for a little longer and see if the final result is what one would expect as shown in Figure 2.2. This phenomenon often happened for slow growing modes.

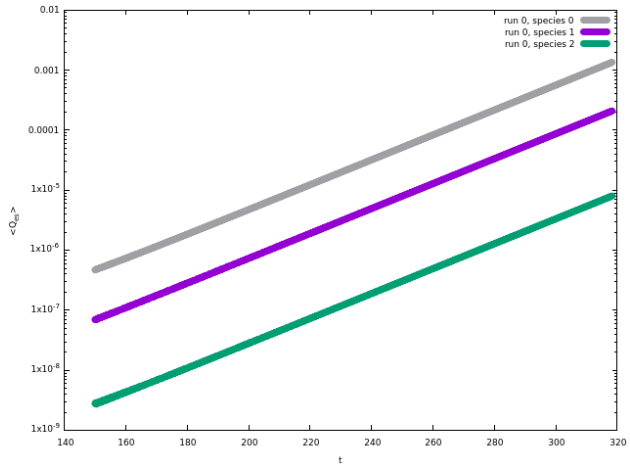

Figure 2.2: GENE global linear simulation gives the expected mode after running for longer time

No growth

High hypz will result stable modes. Hypz for global linear run in pedestal should be less than 10. I am currently using 2.

2.9 Nonlinear Local runs

Nonlinear run is intended to be used as a method for checking the power balance. One need to calculate the τ and Z_{eff} for the local non linear runs is one want to do simulation with 1 specie by runing **"calc zeff id.py"**, the formula is the following

$$\tau = Z_{eff}\frac{T_e}{T_i} \tag{2.9}$$

The common run time is about 36000 cpu hours (960core 40hours) for one species ETG simluation.

The time resolution may influence the result of nonlinear simulation. It may also create numerical instabilities. Please refer to section 3.5 of Merz [1]

To limit the kxmax, kx is calculated

$$kxmax = kxmin * (nx0/2) = (\pi/lx) * (nx0/2) \tag{2.10}$$

Simulation will simulate from $-kxmax$ to $kxmax$

2.9.1 Lilo

```
&general
lilo = T    !Lilo
perf_vec  =  1 1 2 2 1 1 1 1 1
overflow_limit = 80000.0
underflow_limit = 1.0e-20
nonlinear = .T.
x_local = .F.
arakawa_zv = .F.
comp_type = 'IV'
calc_dt = .T.
ntimesteps = 100000
/

&nonlocal_x
rad_bc_type = -1   !For Max
l_buffer_size = 0.1
lcoef_krook = 10.0
lpow_krook = 4
u_buffer_size = 0.1
ucoef_krook = 10.0
upow_krook = 4
drive_buffer = .F.
ck_heat = 0.1
ck_part = 0.1

/

&external_contr
```

```
ExBrate = -1111.0
lilo_w_full_omegator_profile = T   !For Max (allows for variation of shear)
/
```

2.10 Neoclassical Local runs

Historically, neoclassical transport is due the the magnetic curvature. Other transports are contributed by the anomalous transport.

The common run time is about 13000 cpu hours (960core 15hours) for one species neoclassical simluation.

$$\tau = Z_{eff}\frac{T_e}{T_i} \tag{2.11}$$

$$Z_{eff} = (n_i + Z^2 n_z)/n_e \tag{2.12}$$

Z_{eff} is a function of position. One can get the plot of the Z_{eff} by using the script **"calc zeff id.py"**. Can get the value of the T_e and T_i from the local linear runs.

2.11 GENE quantities

2.11.1 Hyp z

Hyper diffusivity is related to upwinding scheme

Hyp stands for hpyer dissution

module load cray-hdf5-parallel
module load cray-petsc-complex
module swap craype-haswell craype-mic-knl

Chapter 3

GENE run's Trouble Shooting

3.1 Common trouble shooting

3.2 Reason of zigzaging

The from the paper mentioned on the last section, In Section 3.1, tt talks about the numerical resaon why there are zigzaging. And how hyper diffusion terms help.

3.3 GENE does not run

GENE complete the job within the seconds and create the folder, then can check the the files:

```
geneerr.log
GENE.#00001.err
GENE.#00001.out
```

Here are some common problems,

3.4 Unmatched processor number

The warning says that:"chosen parallelization is not consistent with the total number of processes".

One need to make sure that the following equation works

```
n_procs_sim*n_parallel_sims=n_procs_s*n_procs_v*n_procs_w*
                            n_procs_x*n_procs_y*n_procs_z
```

One also need to check the submit.cmd file so that
number of node * number of core *64/4= n procs sim*n parallel sims.

3.5 Incorrect geometry namelist for CHEASE

on i/o error: incorrect geometry namelist

For submit file, one need to add another line to let GENE read HDF5.

```
#do not use file locking for hdf5
export HDF5_USE_FILE_LOCKING=FALSE
```

3.6 Re-scaling

GENE will rescal the plot once ϕ grow into a number that is too big to be calculated, this phenomenon is demonstrated by Figure 3.1

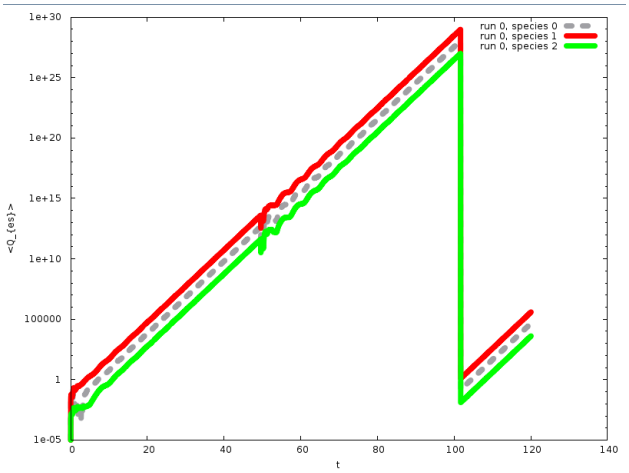

Figure 3.1: GENE re-scale the plot

3.7 "Low time resolution"

With DIIID shot 175823 and following parameter:

x0=0.95, kymin=0.4, coll new = 0.8*coll

this phenomenon is demonstrated by Figure 3.2. The gap is due to negative number of the Q_{es}, so the plot will not show those data points

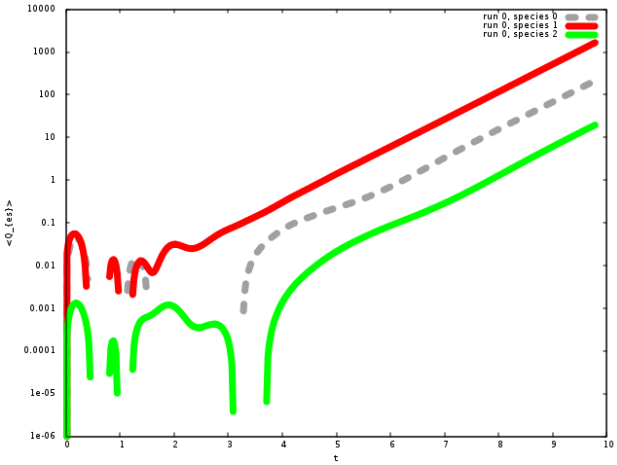

Figure 3.2: GENE low time resolution the plot

3.8 No growth

With DIIID shot 175823 and following parameter:

x0=0.95, kymin=0.11

this phenomenon is demonstrated by Figure 3.3. That plot is showing us a stable mode. Sometime high hypz will resulting in stable mode. For global runs, hyp z should be less than 10. One need to be careful about that.

3.9 Desired mode does not show up

High hpy z might cause certain micro instabilities damped.

Low resolution will fail to capture fine structured micro instabilities.

The following parameter will influence the resolution as well.

edge opt = 2.0

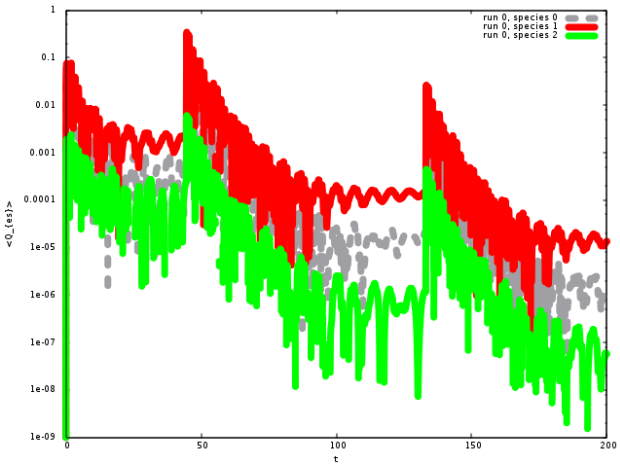

Figure 3.3: GENE no growth plot

3.10 Do not reach time limit

With DIIID shot 175823 and following parameter:

x0=0.98, kymin=0.04

this phenomenon is demonstrated by Figure 3.4. The frequency and the growth rate converged in this plot, so it is fine that GENE does not reach to the time limit.

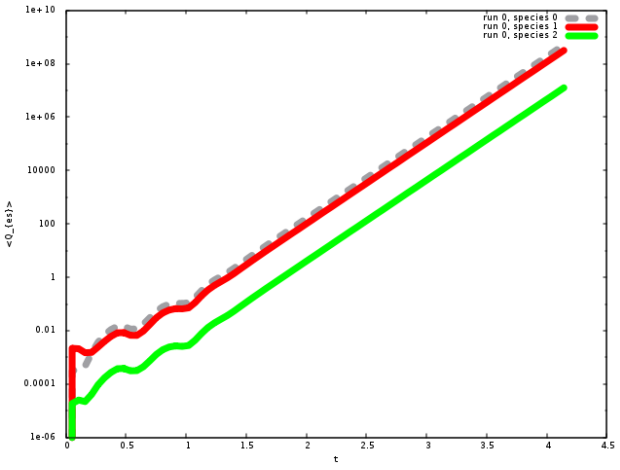

Figure 3.4: GENE Do not reach time limit plot

3.11 Fuzzy plot

With DIIID shot 175823 and following parameter:

x0=0.97, kymin=0.11

this phenomenon is demonstrated by Figure 3.5. The reason is the two mode with very close growth rates and very different frequencies are competing with each other. Here is a quick calculation of this effect.

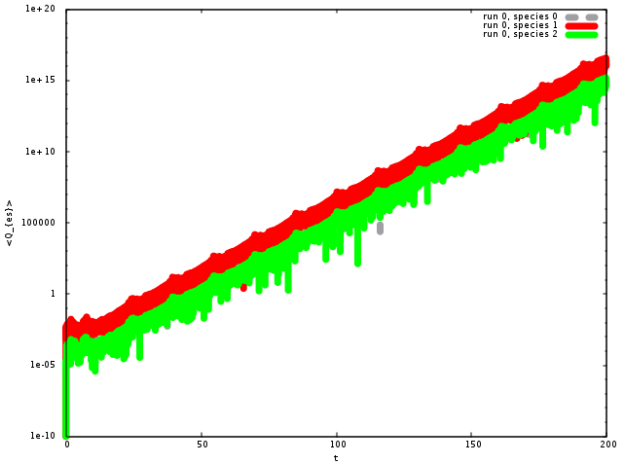

Figure 3.5: GENE Fuzzy plot

$$\begin{aligned}\phi \propto |f|^2 &= |C_1 e^{i\omega_1 t+\gamma_1 t} + C_2 e^{i\omega_2 t+\gamma_2 t}|^2 \\ &= C_1^2 e^{2\gamma_1 t} + C_2^2 e^{2\gamma_2 t} + 2C_1 C_2 cos(\omega_1 - \omega_2) e^{\gamma_1-\gamma_2} \\ &\approx (C_1^2 + C_2^2) e^{2\gamma_1 t} + 2C_1 C_2 cos(\omega_1 - \omega_2)\end{aligned} \tag{3.1}$$

Here is the plot of the equation above in Figure 3.6, the code with be in the supporting file "calc fuzz.nb".

If the run has reach to the time limit, one solution is to increase the time limit (simtimelim in paraemeter file) see if it converges.

3.12 No information of ion and impurity

With DIIID shot 175823 and following parameter:

x0=0.96, kymin=0.13

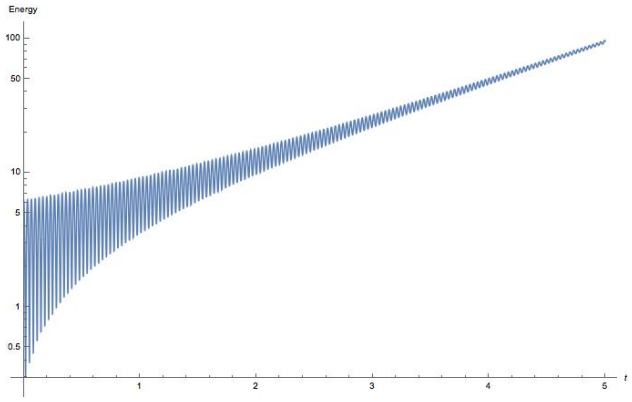

Figure 3.6: Calculated Fuzzy plot

this phenomenon is demonstrated by Figure 3.7. That means the ion and impurity are stabilizing which produce a negative number so the log plot won't show.

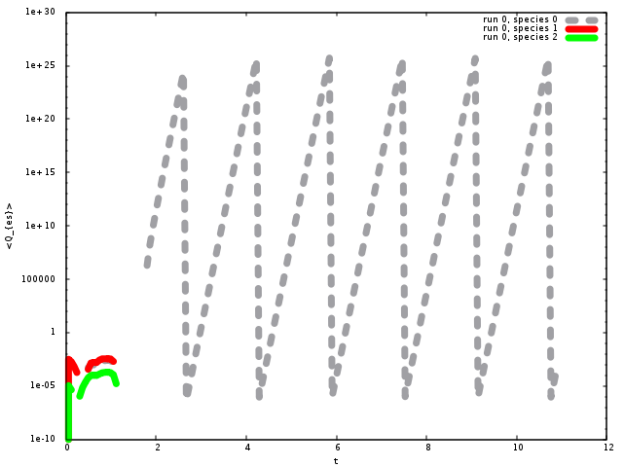

Figure 3.7: GENE No information of ion and impurity

3.13 No growth,really noisy

With DIIID shot 175823 and following parameter:
x0=0.96, kymin=0.15

this phenomenon is demonstrated by Figure 3.8. Similar to the approach to fuzzy.

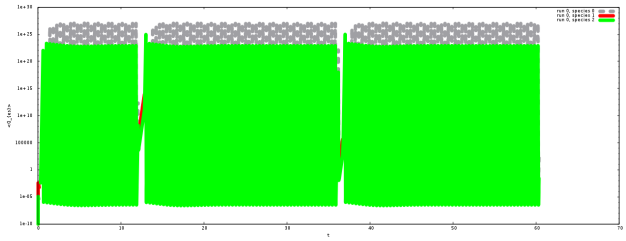

Figure 3.8: GENE No growth, really noisy

3.14 Incorrect general namelist

In global simulation, one need to delete the line which is no longer useful for GENE simulation.

```
coll_f_fm_on = .F.
```

3.15 $\phi - z$ plot is not smooth

From the plotting the (For more detail, please refer to the output processing chapter)

```
scan_info_efit.py
```

One may get a plot that is not smooth (As Figure 3.9 and Figure 3.10 shown), one way to solve this problem is to increase the resolution on z axis. In the "parameter" file, there is a parameter called

```
hyp_z = -8
```

nz0 decrease

Where -8 stands for 10^{-8}, so decrease the this number will increase the resolution of the z axis, hence smooth plot.

One can also increase the absolute value of "hpy z, hyp x, hyp v, nz0,nx0, dt max" in the parameter file to get a better result.

After the adjustment mentioned above, one expect the following result as Figure 3.11 shown.

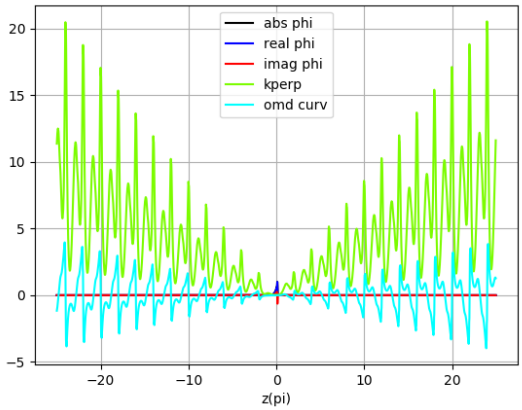

Figure 3.9: Eigenfunction

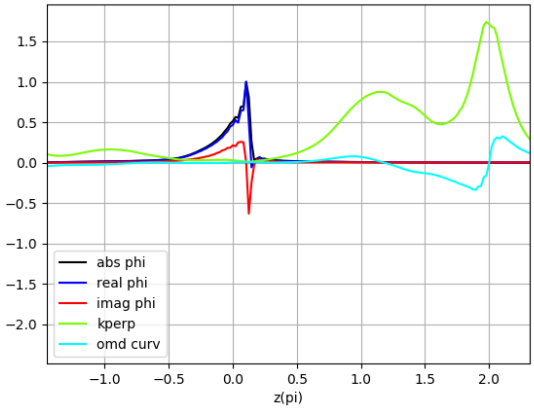

Figure 3.10: Eigenfunction zoom-in

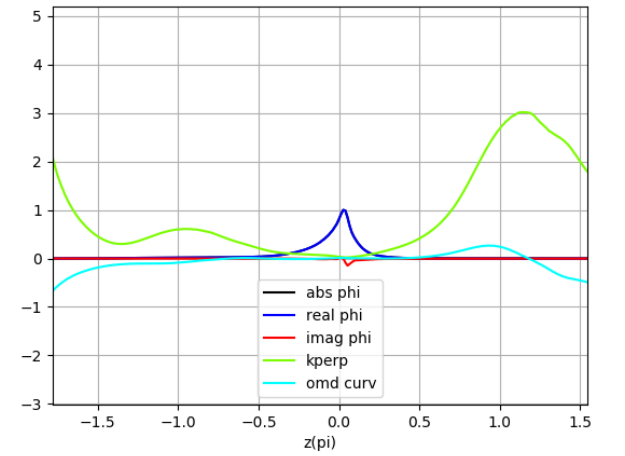

Figure 3.11: Eigenfunction zoom-in smoothed after increase hyp z, nz0 and dt max

Chapter 4

Pre-processing Toolbox

4.1 ITERdb file generator

iterdb file need to be generated from "g" file and "p" file which provide the information for the experiment setup with provide the initial value of the GENE parameter file. the commond is shown below

```
python pwd/make_iterdb.py g0000 p0000
```

Where the number on g and p file is the shot number.

Keep in mind, one could encounter the abnormal profile from reading p file. Since the order and format may differ from shot to shot. One need to modify "make iterdb.py" and "read pfile.py" in order to make it work properly.

For example in case of not having Er, but having vrot

```
vtor = np.empty(0)
dvtor = np.empty(0)
psipvtor = np.empty(0)
#inside of the loop
    n_temp=21
    temp = sdata[(n_temp-1)*nr+i+n_temp].split()
    psipvtor = np.append(psipvtor,float(temp[0]))
    vtor = np.append(vtor,float(temp[1]))
    dvtor = np.append(dvtor,float(temp[2]))
vtor_out = interp(psipvtor,vtor,psi0)
```

```
    dvtor_out = interp(psipvtor,dvtor,psi0)
    return psi0, ne_out, te_out, ni_out, ti_out, nz_out, er_out, vtor_out
```

4.2 ITERdb file plot

One can plot temperature and density of the profile using the following command. But one need to get into the script and change to the correct name of the profile file.

```
plot_profiles_iterdb.py
```

In order to plot q profile, one can try the

```
efit_tools.py g0000000
```

4.3 profile file modification

4.3.1 Constant pressure

One can modify the parameter in the self-consistent way e.g. constant pressure.

First, read into the code and the select the mode, for instance, mode='coll' is to modify the collisionality.

One the change the alpha=target factor to modify the parameter by the factor one set. For expample: if one set

```
alpha=0.8
target_factor=0.8
```

And the original coll=0.8, then the modified coll will be 0.64. while other parameter will be modified as well to the keep the pressure constant.

Second, set the correct file name in "fileName". And the put in the profile files ("profiles e", "profiles i", "profiles z") that "make iterdb.py" have gernerated in profilefilesName. The "base number" is the shoot number. "rhoMidPed" is the "x0" in GENE(Radial distance).

Here is the defintion of "x0" from GENE[2].

this parameter determines which ux surface (labeled by ρ_{tor}, i.e. $0 <$ x0 < 1) will be selected for simulation.

Here is an example of a DIIID shot 175823

```
fileName = 'DIIID175823.iterdb'
profilesName = 'profiles_e'
file_out_base = 'DIIID175823_new.iterdb'
base_number = '175823'
rhotMidPed = 0.95
```

and the change the parameter, then change the iterdb file number and then run the code. The code will take the old iterdb file and gererate a new one, however the iterdb file may have the wrong number,

The command line shown below,

```
python pwd/modProfs_fixedP.py
```

There will be a series of plot to show the information about the new iterdb file and

4.4 CHEASEPY

For more information, please check on CHEASEPY2.0 manual.

CHEASEPY take the iterdb and EQDSK(EFIT/g000000/magnetic geometry)

creat a folder (under the folder shot) name in the format of

```
machine.name shot.Number
e.g. DIIID_162940
```

name the file in such format

```
machine.name_shot.Number_EQDSK
e.g. DIIID_162940_EXPEQ, DIIID_162940_ITERDB, or DIIID_162940_PROFILE
```

create a csv file to include a namelist

There are two ways to run CHEASEPY

```
python runchease.py -m (manual)
python runchease.py -s (submit)
```

Submit GENE run using CHEASE PROFILE

One may need to recompile GENE in order to run CHEASE

The CHEASE will output COCO2. It can be used as geometry file. One can use the COCO file and the profile file to remake the iterdb file and use them as input for the GENE simuliations.

4.4.1 Submit file for CHEASE

For submit file, one need to add another line to let GENE read HDF5.

```
#do not use file locking for hdf5
export HDF5_USE_FILE_LOCKING=FALSE
```

4.5 Extend Buffer

In order to simulate the instabilities that are located around the edge of the device. Using extend buffer can extend the profile over separatrix. Here is the instruction on how to use such script:

Run a global simulation with desired resolution, copy the tracer file and paraemters file into the probe folder.

Use the following command to run the script:

```
python ../../scripts/extend_buffer.py
DIIID175823.iterdb tracer_efit.dat parameters.dat
g175823.04108_257x257 -i
```

-i is the flag for including the impurity.

Change the mag profile into "gene"

Chapter 5

Post-processing Toolbox

5.1 Doppler shift

Local simulation require Doppler shift, Glocal simulation already has Doppler shift.

5.2 General plotting

To plot ϕ VS time, one can use the following line to acheive:

```
$HOME/genecode/tools/gplot nrg_0001
```

To plot A VS time, one need the following line:

```
$HOME/genecode/tools/gplot -n 9 nrg_001
```

5.3 IDL toolbox

A useful reference one can find the thesis of the Dr. Pueschel. Chapter3 is particularly useful.

`http://genecode.org/PAPERS_1/pueschel.pdf`

IDL is a interface base, so it will be really slow when run in the remote SSH(Such as Bitvise SSH Client, PuTT) Nomechain seems to run IDL relatively smoothly.

One need to load a module in order to run IDL toolbox.

```
module load idl
```

Go under the folder diagnostics (/global/homes/m/maxcurie/genecode/diagnostics), then run the following line:

```
idl -vm=vm_diag.sav
```

The interface is shown in the figure 5.1

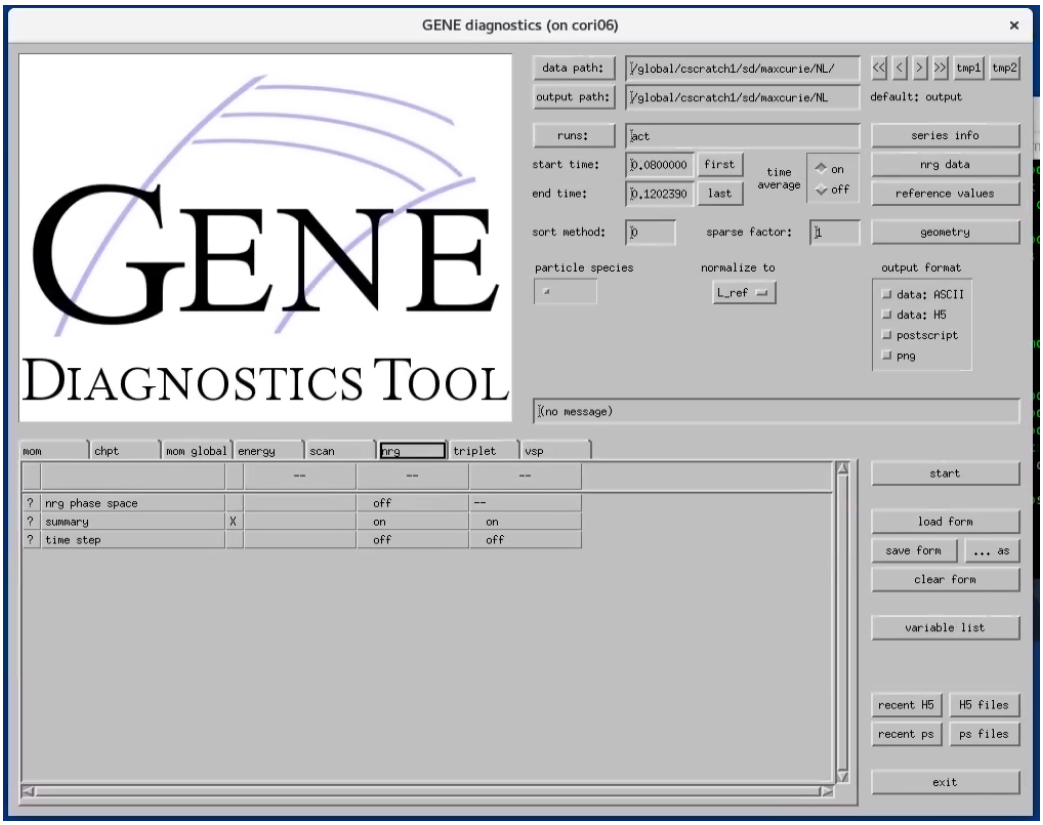

Figure 5.1: The interface of IDL

Type in the path into the data path, and type in the path you want the output to go.

Click run, click on parameter.dat. Click on nrg data. Then you can see the plot of the energy data as shown on the figure 5.2

In order to a report of the whichever information you need, go to the bottom and select the information one needs and click run. Then the following result shown on the IDL, as figure 5.3 shown.

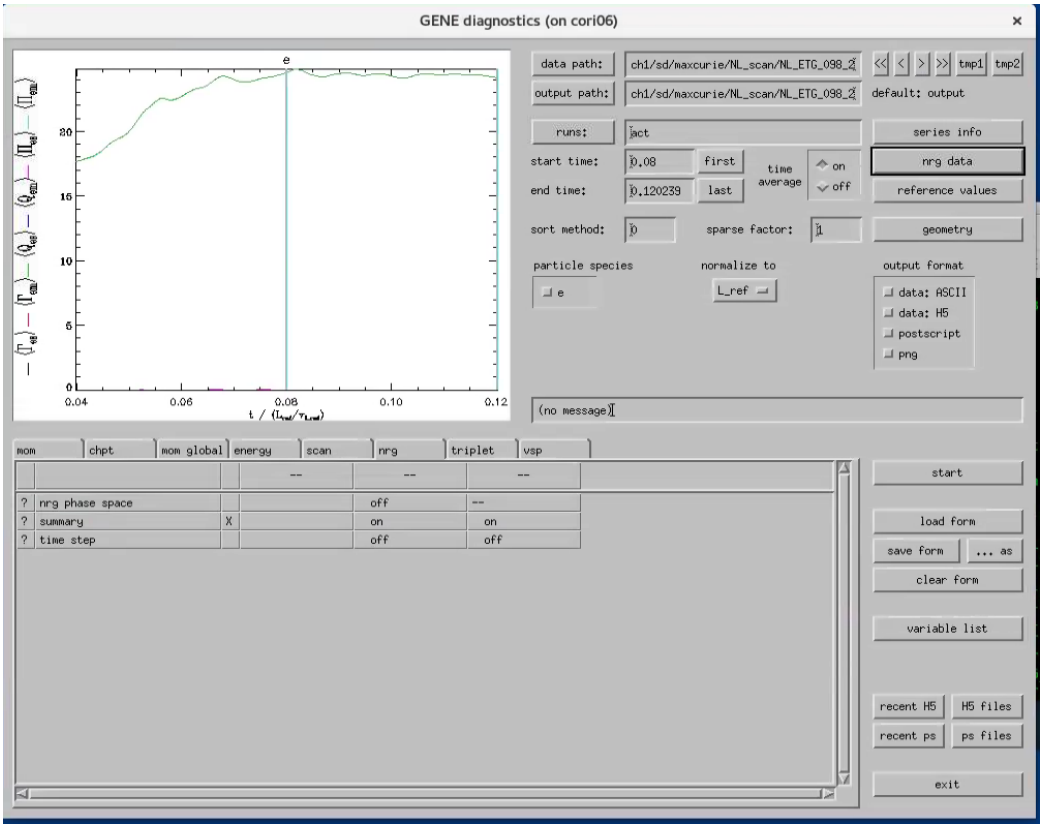

Figure 5.2: The interface of IDL with nrg plto

5.3.1 Increase Resolution

IDL can also increase the resolution of the shot by going under the tab "mom" and chose Toroidal representation. Type in the resolution one want in the "res"

5.3.2 Getting the spectrogram

5.3.3 Modify IDL toolbox

From M.J. Pueschel

"

Almost at the end of diagnostics/prog/frequency.pro, you've got this block:

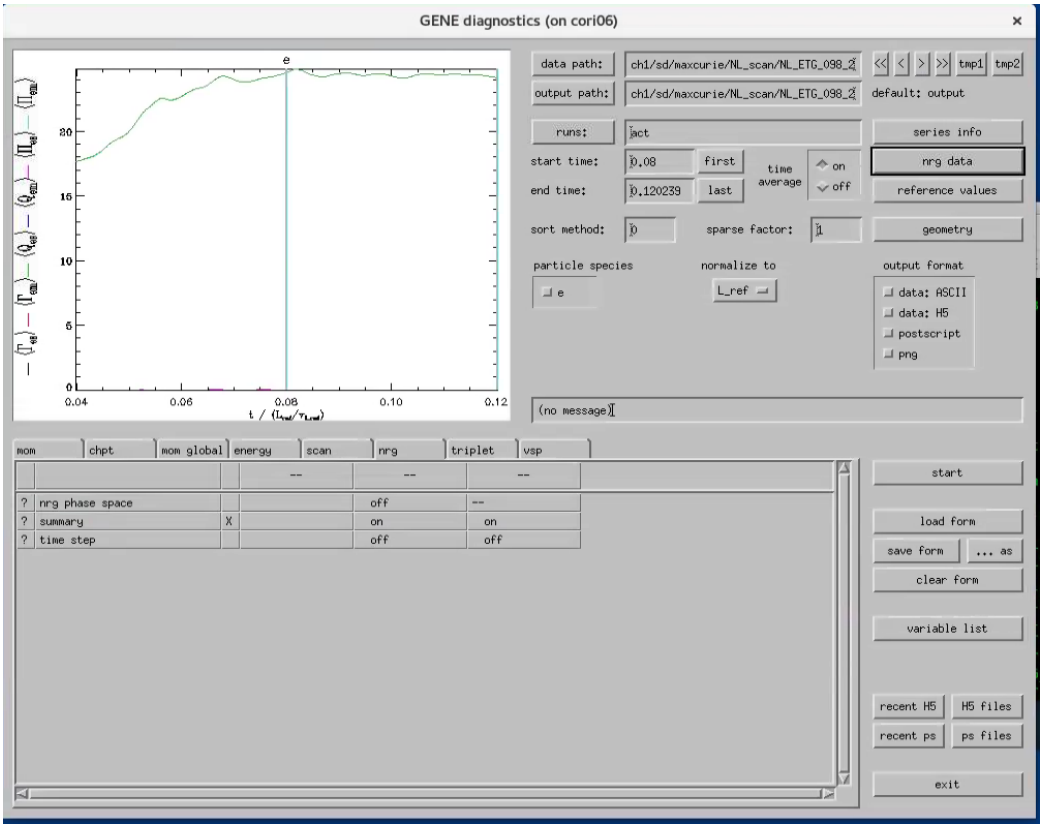

Figure 5.3: The interface of IDL with nrg plto

```
FOR iky = 0, nkyind - 1 DO BEGIN
    win_max = MAX(fftdata_win[ikx,iky,*])
    IF win_max GT 0 THEN fftdata_win[ikx,iky,*] = $
      fftdata_win[ikx,iky,*] / win_max
  ENDFOR
  lev_col = contour_levels([0,1],20)
  CONTOUR, fftdata_win[ikx,*,*], (*series.ky)[(*i).kyind], omega_win, $
    LEVELS=lev_col[*,0], C_COLORS=lev_col[*,1], /FILL,$
    /XSTYLE, /YSTYLE, /XLOG, xrange=[kymin,kymax], XTITLE=xtitle, $
    YTITLE=ytitle, CHARSIZE=csize, CHARTHICK=cthick, YRANGE=yrange, $
    TITLE='!12<!9!!!6'+get_var_string(0,/fancy)+'!9!!!6!U2!N!12>!6'+$
    ((*i).resolve_kx ? ' at k!Dx!N='+$
    rm0es((*series.kx)[(*i).kxind[ikx],0],prec=3) : '')
```

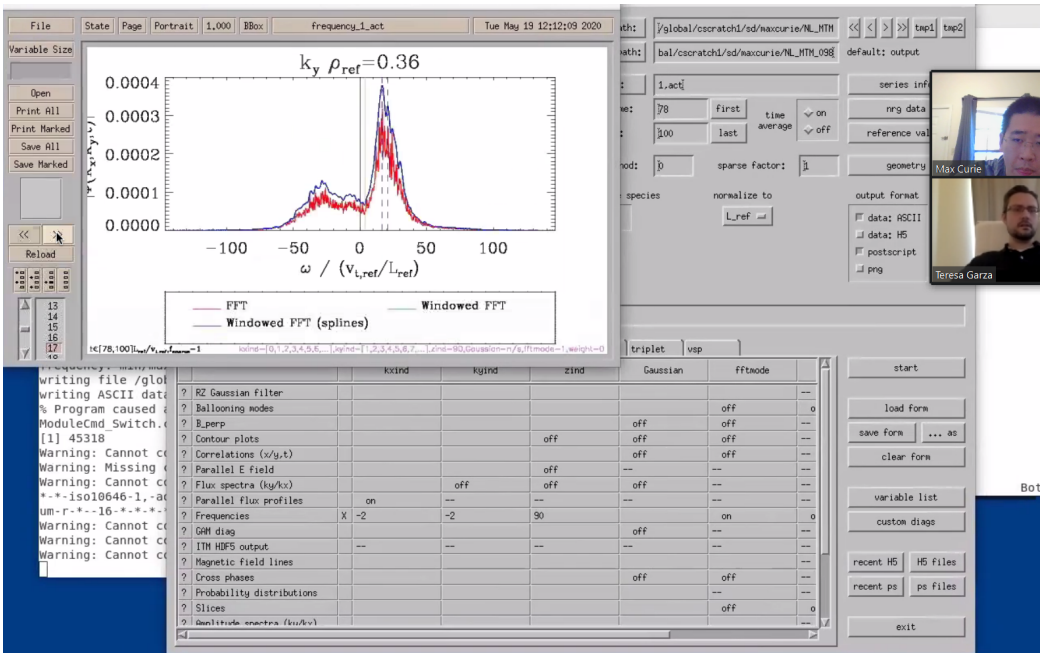

Figure 5.4: The interface of IDL with plotting spectrogram

If you comment out the FOR loop (just at a semicolon at the beginning of each of those five lines), it won't do the separate-k_y normalization, and the FFT already naturally contains the amplitude weight.

To switch from Phi to A_par, at the end of the _init routine, you've got

```
IF par.x_local THEN fft_format, kxky=[0] ELSE fft_format, sxky=[0]
```

There, replace the two zeroes by ones. Then you need to go down a few lines to

```
  IF ((*i).zind EQ -1) THEN data = $
   par.x_local ? (*mom[0,0].kxky)[(*i).kxind,(*i).kyind,*] : $
  (*mom[0,0].sxky)[(*i).kxind,(*i).kyind,*] $
ELSE data = par.x_local ? (*mom[0,0].kxky)[(*i).kxind,(*i).kyind,(*i).zind] : $
  (*mom[0,0].sxky)[(*i).kxind,(*i).kyind,(*i).zind]
```

and replace "*mom[0,0]" by "*mom[0,1]".

Lastly, to output the data into ASCII, you replace the block at the very end

```
 IF NOT (*i).resolve_kx THEN BEGIN
  set_output, diag, header=['ky',rm0es(INDGEN(n_freqs)+1)+$
    '. mode','first moment'], commentline='kxind = '+$
    rm0es((*i).kxind), dat=[[(*series.ky)[(*i).kyind]],$
    [TRANSPOSE(REFORM(omega_max[0,*,0,*]))],[REFORM(omega_moms[0,*])]]
ENDIF
```

by

```
    set_output, diag, dat=fftdata_win[*,*,*]
```

5.4 Neoclassical transport

In order to plot the neoclassical transport one needs:

```
python plot_neoclass.py neoclass.dat
```

If one uses the number as suffix, the following error may appear:

```
Traceback (most recent call last):
  File "/global/homes/m/maxcurie/max/scripts/genediag.py", line 88, in <module>
    units = genetools.units_conversion(paramfpath=paramfpath)
NameError: name 'paramfpath' is not defined
```

One has to use 4 digits format such as "0001"
Using the following line to get the neoclassical transport.

```
python genediag.py -siunits -plotneoclass -display 0001
```

5.5 Fingerprint Method

5.5.1 Transport ratio - D_chi_ratio.py

There are a few parameters that will indicate which mode the plasma is dominating [3]

	Mode Type	χ_i/χ_e	D_e/χ_e	D_z/χ_e	Q_{em}/Q_{es}	Shear-suppressed?
	MTM	~ 0	~ 0	~ 0	> 1	No
	ETG	~ 0	~ 0	~ 0	< 1	No
	MHD-like	~ 1	$\sim 2/3$	$\sim 2/3$	> 1	No
	ITG/TEM	≥ 1	$-0.2 - 1$	~ 1	< 1	Usually

Table 5.1: Theoretical estimates of transport ratio for different instabilities

The MTM(micro-tearing mode) will have the following fingerprint:

$$\frac{\chi_i}{\chi_e} \sim \frac{1}{10} \tag{5.1}$$

$$\frac{D_e}{\chi_e} \sim \frac{1}{10} \tag{5.2}$$

$$\frac{D_Z}{\chi_e} \sim \frac{1}{10} \tag{5.3}$$

The Equation 5.1

```
python $pwd/D_chi_ratio.py #(0001)
```

This read the diffusion coefficient of the last time step

Where #(0001) will be substituted by the file number. The list of parameter will indicate which mode this run is dominating. Q_{EM}/Q_{ES} is the the heat flux of Electromagnetic Versus the heat flux of the ELectrostatic. For MTM, the quantity Q_{EM}/Q_{ES} will be greater than one.

5.5.2 Flucturation ratio - RIP.py

Ratio of fluctuation can be used to indentify mode as well.

For instance,

$$\frac{\delta B/B}{\delta n/n} \tag{5.4}$$

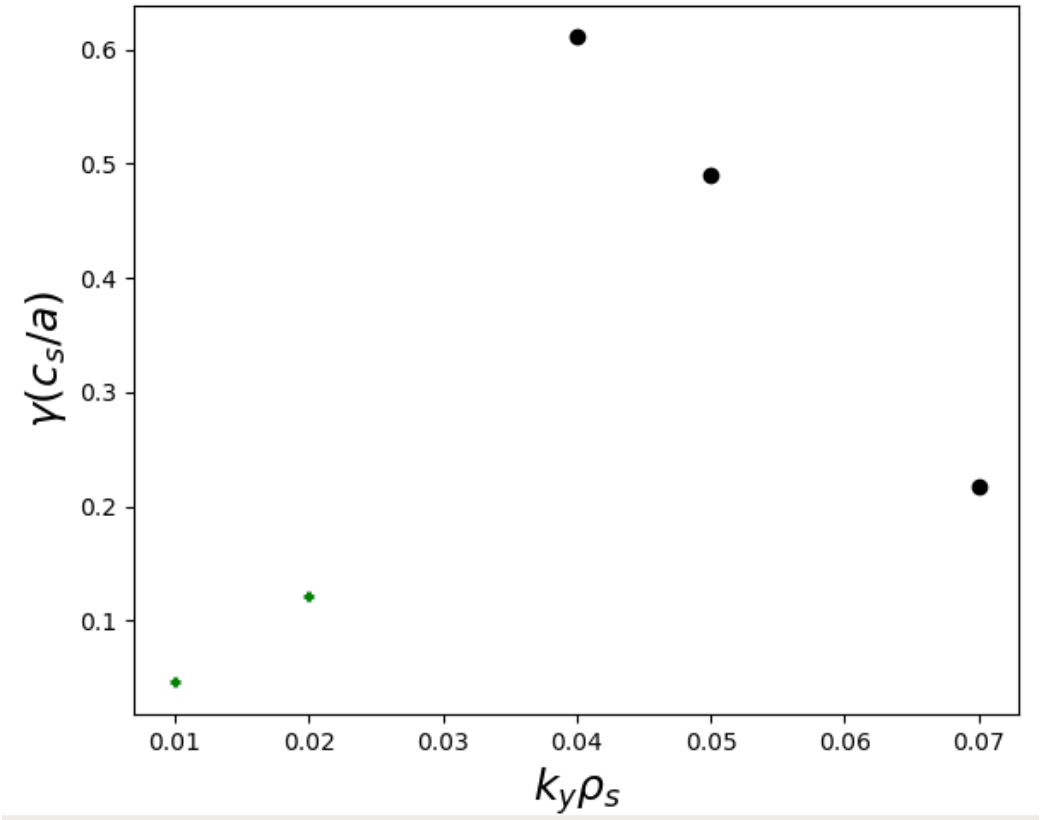

Figure 5.5: Output of the mode finding tool — scan info efit.py

5.5.3 $E_{||}$ - scan_info_efit.py

This method will scan the whole scanfiles, so one need to exit out of the of the scanfiles. The format has to be "scanfiles(name)", for instance "scanfiles0001". The final result is shown as Figure 5.5.

```
python $pwd/scan_info_efit.py #(0001)
```

This script can provide a plot that shows which mode each is.

$$\frac{diff}{abs} = \frac{E_{||}}{k_y|\phi| + \omega|A_{||}|} = \frac{k_y|\phi| - \omega|A_{||}|}{k_y|\phi| + \omega|A_{||}|} \tag{5.5}$$

For the $\frac{diff}{abs}$ is in the different range for different mode

Mode	$\frac{diff}{abs}$	$\frac{Q_{EM}}{Q_{ES}}$	$\frac{\chi_i}{\chi_e}$	$\frac{D_e}{\chi_e}$	$\frac{D_Z}{\chi_e}$
ITG	1	< 1	≥ 1	1	1
ETG	1	< 1	0	0	0
MTM	0.5	> 1	0	0	0
TEM	1	< 1	≥ 1	1	1
MHD	0	< 1	1	$\frac{2}{3}$	$\frac{2}{3}$

For MHD mode, $\frac{diff}{abs}$

5.6 Eigenfunction method

One can also identify different mode by ploting out the eigenmode from from the GENE output the useful commend lines are listed below,

```
python $pwd/plot_mode_structures.py #(0001)
```

For EV(eigenvalue) solver, one can do such command for the mode strucutures.

```
python $pwd/plot_mode_structures.py #(0001) -t #(2)
```

Where the first number is the suffix, second number the the mode one wants to observe.

```
python pwd/xing_TestDrive_fieldsWrapper3.py #(0001) -p T -a T
```

Where $-p$ stands for the ploting the result and "$_a$" stands for writing the output into a output file.

One will have a plot of the region to phi with respect to z, we need to choose the region of z.

The code will also generate some output file in the form of the following

```
avg3_kz_kperp_omd_0001
```

It provides the information about the k and frequency and growth rate.

The MTM will be have a relatively low amplitude at z=0, on the other hand TEG is will have large amplitude at z=0.

5.6.1 Eigenfunction Calculator

5.7 field grwoth rate calculator

5.7.1 Manual method

Using the following python script, one can calculate the growth and frequency from the field by manually selecting the range of the

```
calc_omega_from_field.py
```

5.7.2 Automatic method

Using the following python script, one can calculate the growth and frequency from the field by automatically selecting the range from a set a critiria that determine which segment is the stedy growth state.

```
omega_scan_max.py
```

One can change the criteria of the calculation, in the file "omega find line,py", one can change the following 3 parameter,

1. binsize
   This parameter allow user to change the tolerant error, the initial value is 0.02

2. space
   This parameter allow user to change the spacing between the line (1/density of the dots), the initial value is 3

3. length
   This parameter allow user to indicate the minimum length of the line, the initial value is 15

5.8 Compare with experiment data

5.8.1 BES(Beam Emission Spectroscopy)

This script can produce the output that can be compared with the experiment data in BES

```
BES_new.py
```

5.8.2 SI unit of frequency

This script can produce the output frequency and grwoth rate in SI unit so that one can compare it with Experiment data

```
omega_gene_to_hz.py
```

5.9 Global run

In order to use the script, one need to transform the output format that works for those scripts, one need to type in the following line:

```
$HOME/genecode/tools/runassign 1
```

Then the number of the run will be 1 (compared with 0001 in local runs:)

5.9.1 eigenfunction finding

```
plot_mod_structures.py 1
```

It will provide heat plot with x axis being minor radius and y axis

```
plot_E_t nrg_1 -n 9
```

5.9.2 D chi

```
D_chi.py
```

Plot spatially average QEM QES

5.10 Doppler shift

Doppler shift maybe needed for the consistancy with experimental measurement.

5.11 Tool boxes

read_write_geometry.py

Chapter 6

Construct One's own Toolbox

Eventually, one will need to construct a script that can output a quantities that can be compared with Experiment or Analytical results. Which chapter is designated for guide you through the procedure on construction of your own script with minimum effort thank to the existent script that does most of the job for us.

6.1 Pre-processing

6.1.1 List of equilibrium data file

One can give all the equilibrium profile from p file(profile file) and g file(geometry file). Sometime. p file is not well organized. One can use 'make_iterdb.py' to read in g and p file to and output ".iterdb", "profile_e", "profile_i", and "profile_z". In summary, one may have the following file in the prob folder

- g000000.00000
- p000000.00000
- 0000000.iterdb
- profile_e
- profile_i
- profile_z

There is a naming convention: for p and g file one may see: g175823.04108, where"175823" is the shot number, "04108" is the time stamp in ms. So "**g175823.04108**" can be translated into English "**geometry file of shot 175823 at 4108 millisecond**"

6.1.2 ITERDB file

T is in unit of eV.

6.2 Basic structure of the GENE output

Check on GENE manual chapter4 for more information.

δ_n will be provided in the mom file. the global linear simulation output has the following dimension: (z,ky,x).

$A_{||}$ can extracted from field file. The dimension is (z,ky,x)

6.2.1 zgrid

For location simulation, ballooning angle is discretized in such way

6.2.2 xgrid

For global simulation, radius has been discretized in such way.

```
if 'lx_a' in pars:
    xgrid = np.arange(field.nx)/float(field.nx-1)*pars['lx_a']+pars['x0']-pars
else:
    xgrid = np.arange(field.nx)/float(field.nx-1)*pars['lx'] - pars['lx']/2.0
```

6.3 GENE unit-SI unit-Gauss unit

Field wrapper is the script that construct that one can to read the field file that GENE produces.

For instance, "eigenfunctions from field file" can provide ϕ and $A_{||}$

Get to the real space from read write geomtry.py Use geo R and geo z

6.4 Some Usefule lines

Bibliography

[1]V. Von, F. Merz, and A. Rottweil, "Gyrokinetic simulation of multimode plasma turbulence", (2008) http://hdl.handle.net/11858/00-001M-0000-0026-FDD2-1.

[2]G. D. Team, "The gyrokinetic plasma turbulence code gene: user manual", (2018).

[3]M. Kotschenreuther and et al., "Gyrokinetic analysis and simulation of pedestals, to identify the culprits for energy losses using "fingerprints"", 5–6 (2017).