A brief introduction to GENE

Gene Development Team

support@genecode.org

December 18, 2018

WARNING

This tutorial aims at providing the basics for installing and running GENE. Please check

- the website `http://genecode.org`
- the documentation

for more recent or more detailed information regarding required libraries, input parameters, file formats etc.!

Outline

What is GENE ?

GENE = **G**yrokinetic **E**lectromagnetic **N**umerical **E**xperiment

- designed for numerical investigations of plasma microturbulence
- solving the δf-splitted gyrokinetic system of equations using a Eulerian approach (fixed grid) in 5D phase space
- initial value or eigenvalue computations
- radially local[a] (flux tube) or global (up to full torus) simulation domain
- various interfaces to MHD equilibrium codes and transport solvers
- massively parallelized using (mainly) MPI and OpenMP

[a] addressed in this tutorial

Gyrokinetic system of equations I

Gyrokinetic Vlasov equation per species σ with collisions

$$\frac{\partial f_\sigma}{\partial t} + \dot{\mathbf{X}} \cdot \nabla f_\sigma + \dot{v}_\parallel \frac{\partial f_\sigma}{\partial v_\parallel} + \dot{\mu} \frac{\partial f_\sigma}{\partial \mu} = C(f_\sigma, f_{\sigma'}) \tag{1}$$

gyrocenter position $\mathbf{X}$

$$\dot{\mathbf{X}} = v_\parallel \mathbf{b}_0 + \frac{B_0}{B_{0\parallel}^*} \mathbf{v}_\perp$$

with the combined drift velocities

$$\mathbf{v}_\perp \equiv \frac{c}{B_0^2} \bar{\chi}_1 \times \mathbf{B}_0 + \frac{\mu}{m_\sigma \Omega_\sigma} \mathbf{b}_0 \times \nabla B_0 + \frac{v_\parallel^2}{\Omega_\sigma} (\nabla \times \mathbf{b})_\perp$$

and the generalized potential
$\bar{\chi} = \bar{\phi}_1 - \frac{v_\parallel}{c} \bar{A}_{1\parallel} + \frac{\mu}{q_\sigma} \bar{B}_{1\parallel}$

parallel velocity $v_\parallel$

$$\dot{v}_\parallel = \frac{\dot{\mathbf{X}}}{m_\sigma v_\parallel} \cdot \left(q_\sigma \bar{\mathbf{E}}_1 - \mu \nabla (B_0 + \bar{B}_{1\parallel}) \right)$$

with the electric field
$\mathbf{E}_1 = -\nabla \phi_1 - \frac{\mathbf{b}_0}{c} \frac{\partial}{\partial t} A_{1\parallel}$

magnetic moment μ

$$\dot{\mu} = 0$$

... which is δf-splitted in GENE

Gyrokinetic system of equations II

Gyrokinetic field equations

Poisson equation

$$\nabla_\perp^2 \phi_1 = -4\pi \sum_\sigma q_\sigma n_{1\sigma}$$

Ampère's law

$$\nabla_\perp^2 A_{1\parallel} = -\frac{4\pi}{c} \sum_\sigma j_{1\parallel\sigma}$$

$$B_{1\parallel} = -4\pi \sum_\sigma \frac{p_{1\perp,\sigma}}{B_0}$$

. . . all in all a nonlinear, 5D partial integro-differential system of equations

Relevant moments (in local approx./Fourier space)

$$n_{1\sigma,\mathbf{k}} = \frac{2\pi B_0}{m_\sigma} \int \mathrm{d}v_\parallel \mathrm{d}\mu \left[J_0 h_{1\sigma,\mathbf{k}} - q_\sigma \phi_{1,\mathbf{k}} \frac{F_{0\sigma}}{T_{0\sigma}} \right]$$

$$j_{1\parallel\sigma,\mathbf{k}} = q_\sigma \frac{2\pi B_0}{m_\sigma} \int \mathrm{d}v_\parallel \mathrm{d}\mu \; v_\parallel \left[J_0 h_{1\sigma,\mathbf{k}} - q_\sigma \phi_{1,\mathbf{k}} \frac{F_{0\sigma}}{T_{0\sigma}} \right]$$

$$p_{1\perp\sigma,\mathbf{k}} \equiv \frac{2\pi B_0}{m_\sigma} \int \mathrm{d}v_\parallel \mathrm{d}\mu \; \mu B_0 I_1 h_{1\sigma,\mathbf{k}}$$

with the nonadiabatic part of f_1

$$h_{1\sigma} \equiv f_{1\sigma} + \left[q_\sigma J_0 \phi_1 + \mu I_1 B_{1\parallel} \right] \frac{F_{0\sigma}}{T_{0\sigma}}$$

and the Bessel functions

$$J_0 = J_0(k_\perp \rho)$$

$$I_1 = I_1(k_\perp \rho) = 2J_1(k_\perp \rho)/(k_\perp \rho)$$

How to get and install GENE

How to get and compile the code

1. visit `http://genecode.org`,
   fill and submit the short form in “Get GENE”
2. download the code as indicated via email

   ```
   git clone https://<USER>@gitta.rzg.mpg.de/~GENE/guest/git.py/gene.git -b release-<X.Y> <genedir>
   ```
3. change to the <genedir> directory
4. call `make doc` or `gmake doc` to create the documentation to be found in <genedir>/doc/gene.pdf
5. call `make mach_wrapper` and check resulting string <mach>
   - if <mach> is other than `new_machine` and clearly related to your machine, **check the header of makefiles/<mach>/<mach>.mk for modules to be loaded – on some machines you may also run** `make get_required_modules` (skip the next slide)
   - otherwise, you need to set up the machine dependent part of the makefile (see next slide)
6. finally, type `make` (or `make -j` for parallel compilation); the resulting binary can be found in <genedir>/bin/gene_<mach>

Setting up the makefile for an unknown machine

- define a meaningful name for your machine (`<mach>` in the following)
  - either by adding an appropriate identification in `<genedir>/makefiles/machine.def` (preferred; *one time change only*)
  - or by *permanently* setting the environment variable `MACHINE`
- copy and rename the makefile template from `<genedir>/makefiles/new_machine/new_machine.mk` to `<genedir>/makefiles/<mach>/<mach>.mk`
- open `<genedir>/makefiles/<mach>/<mach>.mk`, search for and fill at least the following tokens (check the documentation for details)
  - the Fortran compiler `COMPILER` (e.g., intel, gnu, cray, pgi, etc.)
  - `FFTLIB` (fftw, mkl or essl) and the corresponding block
  - the compiler flag for double precision (search for `DOUBLE_PREC`)
  - the **complex** versions of PETSc and SLEPc via `PETSC_DIR`, `SLEPC_DIR`, `PETSC_ARCH`; set `SLEPC = yes`
- compile, run the *test suite* and optionally send modifications to support@genecode.org

note: changes to `makefiles/<mach>/<mach.mk>` require `gmake distclean`

Setting up a simulation

Running GENE

Example

This would be the way how to call GENE in a shell or batch script with 8 MPI processes
`mpiexec -n 8 ./gene_<mach>`
where `mpiexec -n` might need to be substituted by *mpirun, aprun, poe, ...*

The input file

The simulation parameters are set in the `parameters` input file which ***must be located in the same directory*** & will be ***read upon simulation start***

Remarks

- a strategy to organize your simulations is to create individual `prob` ($\sim$ problem) directories containing `parameters`, a link to the executable, batch scripts, etc. for each of them
- the command `./newprob` in `<genedir>` will automatically perform those steps

The `parameters` input file

Setting up the `parameters` input file is the crucial part when attempting to run a simulation. As this file uses formatted data and Fortran namelists, it can either be modified with

(a) the *text editor* of your choice

or

(b) the *graphical launcher*
... which furthermore tries to automatically resolve inconsistencies and assist with job submission

The `parameters` file - parallelization namelist

```
&parallelization
n_procs_s =   1   !divisor of n_spec
n_procs_v =   1   !divisor of nv0 (max: 0.5*nv0)
n_procs_w =   8   !divisor of nw0
n_procs_y =   1   !divisor of nky0 (keep low)
n_procs_z =   1   !divisor of nz0 (maximum: 0.5*nz0)
/
```

- This namelist sets the number of **MPI** processes in each dimension.
- Set to ≤ 0 if the appropriate value is unknown and should be detected by GENE (will increase the initialization time); Lower (upper) bounds for the automatic parallelization tests can be specified by `min_npA` (`max_npA`) where `A=s,v,w,y` or `z`.

The `parameters` file - box namelist I

Perpendicular box sizes

```
&box
kymin    = 0.05 ! min. binormal Fourier mode
adapt_lx =  .t. ! adapt lx to to ky mode (in linear sim's)
lx       = 128. ! radial box width (determines Δk_x)
```

- typical `kymin` value $\lesssim 0.1$ (ly $\gtrsim 60\,\rho_{\mathrm{ref}}$) for nonlinear sim's
- `adapt_lx = .t.` **replaces** lx by 1/(ky·shat) to maximize number of poloidal connections `nexc` in **linear** sim's
- `lx` should be $\gtrsim 60\,\rho_{\mathrm{ref}}$ and will be adjusted to kymin · shat · lx $\in \mathbb{N}$; alternatively, the number of poloidal connections `nexc` can be specified

```
kx_center = 0.0 ! kx center value for linear studies
```

- typically, linear investigations are considering the dominant nonlinear mode which is typically $k_x = 0$(+connections); however, particularly in strongly shaped geometries other k_x values might be interesting, as well.

The `parameters` file - box namelist II

Perpendicular grid points

```
nx0 = 128   !number of grid points in x direction
nky0 = 16   !number of Fourier modes in y direction
```

guideline for perpendicular grid points:

- linearly:
  - at least `nx0` $\gtrsim 4$ (if `adapt_lx` is true)
  - `nky0` $= 1$ - individual simulations per `ky` are recommended!
  - $k_y =$`kymin` (and multiples if `nky0`> 1)
- nonlinearly:
  - radial resolution `lx`$/$`nx0` < 1 (down to $\mathcal{O}(0.1)$);
  - number of Fourier modes in binormal (y) direction should be `kymin` $\cdot$ `nky0` $\gtrsim 1.0$ (check for vanishing growth rate with linear sim's);
    powers of low prime numbers are recommended (better FFT performance)
  - $k_y = 0,$ `kymin`$, .., $`kymin` $\cdot$ (`nky0` $- 1$)

remark: number of y grid points in real space `ny0` $= 2$ `nky0`

The `parameters` file - box namelist III

Velocity space grid

```
lv =   3.0   ! box size in v_|| direction
lw =   9.0   ! upper simulation box end in μ direction
```

- lv should be ≥ 3.0 (in units of $v_{T\sigma} = (\mathbf{2}T_{\sigma 0}/m_\sigma)^{1/2}$, σ : species index)
- the parallel velocity points span the range from $-v_{T\sigma}$ to $v_{T\sigma}$
- lw should be ≥ 9.0 (in units of $T_{\sigma 0}/B_{\mathrm{ref}}$)

```
nv0 =  32   !v_|| grid points (usually 32, 64)
nw0 =   8   !μ grid points (typically 8, 16)
```

- equidistant grid in $v_\parallel$, Gauss-Laguerre knots for μ grid

The `parameters` file - box namelist IV

```
nz0 =  16  !grid points in z (||) direction
```

- simple circular geometries: 16 or 24 parallel grid points sufficient
- realistic MHD-equlibria: up to 100 grid points needed

```
n_spec = 1 !number of particle species
```

- n_spec = 1: oppositely charged species is assumed to be adiabatic!

```
x0 = 0.5 !macroscopic radial position of box center in L_ref
```

- req'd only if some parameters are set by MHD or profiles interfaces; often given as normalized ρ_{tor}

The `parameters` file - in_out namelist

Usually, no changes are needed in this namelist except for the following parameters:

```
&in_out
diagdir = '/ptmp/exampledir'   ! output file dir
read_checkpoint = .f.
```

- output files (e.g. nrg, mom, field) are written to `diagdir`; make sure to use a high performance I/O file system
- to continue an old simulation set `read_checkpoint` to `.t.` to read an existing `checkpoint` file from `chptdir` (=`diagdir` by default) when starting the next simulation

The `parameters` file - general namelist I

```
&general nonlinear = .t.     !switch on/off nonlinear terms
comp_type = 'IV'    !initial(IV) or eigenvalue(EV) comput.
```

`comp_type = 'IV'`

```
calc_dt = .t.      !switch on/off
                   !dt_max calculation
dt_max  = 0.0385   !maximum time step
                   !in L_ref/c_ref
```

`comp_type = 'EV'` (lin. only)

```
n_ev = 1 !no. of EVs
```

- calc_dt = .t.
  - PETSc/SLEPc linked → exact determination of `dt_max`
  - not linked → approximated by CFL
- calc_dt = .f.
  - use `dt_max`

The `parameters` file - general namelist II

Termination conditions

```
ntimesteps =  100000 !maximum number of time steps
timelim    =   64500 !(system) time limit in sec
simtimelim =    5000 !simulation time limit
omega_prec =    1e-3 !convergence precision
```

- simulation will stop in a controlled way, if
  - number of time steps exceeds `ntimesteps`
  - simulation runs longer than `timelim` seconds; set $<$ wallclock limit!
  - simulation time in $L_{\mathrm{ref}}/c_{\mathrm{ref}}$ reaches `simtimelim`
  - growth rates and real frequencies are converged up to a precision of `omega_prec` (only for linear simulations and $\mathrm{istep\_omega} > 0$).

The `parameters` file - general namelist III

Physics parameters

```
beta         =  0.71E-02 !β_ref ≈ 403 · 10^-5 n_e19 T_ref,keV / B^2_ref,T
coll         =  0.68E-02 !normalized collision frequency
collision_op = 'landau'  !'landau' or 'none'
debye2       =  0.0      !normalized λ^2_Debye / ρ^2_ref
```

Hyper diffusion

```
hyp_x =   0.0   !radial hyper diffusion (keep small)
GyroLES = T     !nl. sims only! adaptive perp. hyper diffusion
hyp_z =   2.0   !parallel hyper diffusion
hyp_v =   0.2   !v_|| hyper diffusion
```

- If parallel profiles show *zig-zag* patterns and `nx0` and/or `nz0` cannot be increased try `hyp_z = -1` for automatic adjustment
- to avoid recurrence effects, we recommend to set `hyp_v` to a value $\sim 0.1 - 0.5$ in **collisionless** simulations and to 0 or small values else

The parameters file - geometry namelist

Geometry

```
&geometry
magn_geometry = 's_alpha'  !'slab','circular','miller'
                           !'chease','tracer_efit','gist'
```

analytic models: s_alpha, circular, miller, slab

```
q0        =  1.4    !safety factor q
shat      =  0.796  !magnetic shear parameter
amhd      =  0.0    !α_MHD = −q²R(dβ/dr)  (-1 for automatic)
trpeps    =  0.18   !r/R  at flux tube position r
major_R   =  1.0    !major radius with respect to L_ref
```

interfaces: chease, tracer_efit, gist

minimum requirement:

```
geomfile  =  'examplefile' !declaration of interface file
```

Optional: enforce plasma current (Ip) / toroidal magnetic field (Bt) orientation (see documentation)

```
sign_Ip_CW = -1   !counter-/clockwise (-1/1)
sign_Bt_CW =  1   !counter-/clockwise (-1/1)
```

The `parameters` file - species namelist(**s**)

Add a species namelist for each species:

```
&species
name = 'ions'   !species name string
omn  =   2.22   !density gradient ω_n = −(L_ref/n) (dn/dρ)
omt  =   6.92   !temperature gradient ω_t = −(L_ref/T) (dT/dρ)
mass =   1.0    !mass normalized to m_ref
charge = 1      !charge norm. to elementary charge
temp =   1.0    !temperature normalized to T_ref
dens =   1.0    !density normalized to n_ref
/
```

- check documentation for interfaces to profile files (e.g., `prof_type` `iterdb_file`, etc.)

The `parameters` file - units namelist

Optional(!) namelist for

- denormalisation in post-processing routines
- consistent and automatic calculation of physics parameters as, e.g., β

```
&units
Tref = 0.9688   !reference temperature in keV
nref = 4.081    !reference density in 10^19 m^-3
mref = 2.0      !reference mass in proton mass units
Lref = 0.6757   !reference length in m
Bref = 2.424    !reference magnetic field (typically on axis) in T
omegatorref = 6305 !reference toroidal angular velocity in rad/s
/
```

- check documentation for automatic calculation in profile file interface cases

The `parameters` file - automatic computations

Recommended/helpful settings if all ref. value are filled/requested(*) in `units`:

```
&box
n0_global = -1111 !find min. toroidal mode number closest
                  !to given kymin or force a given value
adapt_ly = T !adapt kymin to n0_global for given rhostar;
             !recommended for applications to expt.
&general
beta = -1     !beta consistently from ref. values
coll = -1     !coll consistently from ref. values
debye2 = -1   !debye2 consistently from ref. values
zeff = -1     !zeff from data base(*)

&geometry
amhd = -1     !amhd consistently from gradients and beta
rhostar = -1 !rhostar consistently from ref. values
dpdx_pm = -2 !pressure gradient from equilibrium (amhd)

&external_contr
ExBrate = -1111 !ext. shear flow rate from data base(*)
pfsrate = -1111 !parallel flow shear rate from data base(*)
```

(*) requires active data base interface (see documentation/`prof_type`)

The `parameters` file

Some remarks

- Information on further parameters, e.g. concerning the initial condition, numerical schemes, frequency of data ouput, etc. can be found in the GENE documentation located in the `/doc/` subdirectory of your GENE installation.

Convergence

- In principle, each simulation has to be carefully checked for convergence with regard to numerical parameters that is e.g. doubling box sizes and/or grid resolution for testing purposes.

Setting `parameters` with the launcher I

- start python based GUI by calling `./GENE-GUI.py` in `<genedir>`
- click on `Default values` to fill with Cyclone-like parameters
- set a least the number of total MPI processes and check all parallelization choices if not sure what to do here
- click on `New 'prob' dir.` for creating a new problem directory
- . . . and finally save your parameters here

Setting `parameters` with the launcher II

- fill at least the Output directory

Setting `parameters` with the launcher III

- fill proper grid numbers (rules of thumb in previous section)
- note that the number of poloidal connections (`nexc`) can be entered directly and consistently with the radial box extension (`lx`)

Setting `parameters` with the launcher IV

- set time limit according to your batch system
- fill dimensionless physical parameters as needed

Setting `parameters` with the launcher V

- choose MHD equilibrium
- fill the requested fields

Setting `parameters` with the launcher VI

- add and remove species using the 'New species' and 'Delete' buttons
- fill the corresponding list for each species
- to read numerical profiles select the appropriate profile type, and fill the requested fields

Setting `parameters` with the launcher VII

- optionally fill reference values for denormalization in the post-processing
- optionally set dependent dimensionless input parameters self-consistently by clicking the buttons

Setting `parameters` with the launcher VIII

- Check
- and Submit
  (this requires a `launcher.cmd` batch script template in `makefiles/<mach>/`)

The `scanscript`

A well known problem ...

The need to scan over multiple parameter values, e.g., to determine a threshold

Solution

- add Fortran comments as follows to the desired parameters, e.g.,
  `omt = 6.9 !scanlist:0.0,3.0,4.0,5.0,6.9`
  ...representing a scan over a list of temperature gradient values
  This also works in the launcher tool!
- use the proper system call!
  - either just activate *parameter scan* in the launcher
  - or replace `mpiexec -n <NMPI> ./gene_<mach>` in your batch script by
    `./scanscript -np <NMPI> -ppn <MINPROCS>`
    `<NMPI>`: total number of mpi processes
    `<MINPROCS>`: minimum number of processors per simulation (scanscript might run several simulations in parallel for optimum efficiency)
- (linear) scan results will be collected in `diagdir/scanfiles<ID>/scan.log` at the *end of the job*; call `./scanscript --mks` in the directory where the job has been launched for a manual update of `scan.log` at runtime
- check `./scanscript --help` and the documentation for further options and output file arrangement

Analyzing and post-processing

The Diagnostics Tool

The Diagnostics Tool I

Starting

- regular IDL version (requires IDL licence):
  full functionality, user can customize diagnostics
- IDL Virtual machine version:
  no manipulation of source possible; IDL VM is free
- Both versions: platform-independent (Unix, Mac OS, Win32; for latter, see documentation)

- IDL: start `idl` in diagnostics directory, then run `@diag`
- IDL VM: start `idl -vm=vm_diag.sav`

Warning

The diagnostics tool needs to be run on a platform from which one's data is accessible via the file system!

The Diagnostics Tool II
Graphical User Interface

- `runs`: e.g. `256,261-264,270+, [enter]` rec'd
- `series info`: important parameters → window
- `nrg data`: nrg time traces → window
- `nrg diag`: growth rate and saturation analysis
- `geometry`: parallel variation → window

- `diag table`:
  - tabs: analyze mom/scan/vsp files
  - `?`: help text
  - activation status: if active, `X`
- `save and start`: selected mom/scan/vsp diags
- `variable list`: variable $\leftrightarrow$ number for diag table par's
- `recent ps`: ps files in `gv` after `save and start`
- `ps files`: open dialog, output directory
- `exit`: close, GUI state not saved

The Diagnostics Tool III

Using the Diagnostics Tool

- preliminaries (to check run parameters and/or time evolution): run number, `[enter]`, `normalization`, `show series info`/`show nrg data`
- full `nrg` analysis: time window, `species`, `normalization`, `output format`, `start nrg diag`, (hourglass cursor), `recent ps`
- full `mom`/`scan`/`vsp` analysis: time window, `species`, `normalization`, `output format`, select diags in table (check third column with `X`), fill in diag parameters, repeat this with other diags, `save and start`, wait for completion dialog, `recent ps`

The Diagnostics Tool III

Using the Diagnostics Tool

- for help with specific diags, press `?` next to diag name → general description, parameters info
- custom diags: can modify with Diagnostics Tool open, auto-recompiled on `save and start` (only with full IDL)

The Diagnostics Tool IV

Scans and Data Output

Scan analysis: `runs` – ignored; `sort method` – `0`
Data output:

- (plain text) data files with predefined output
- for modification/specific info, change source: `prog/[diag].pro`
  → search for `dat=`
  → modify content (e.g., variables) to be written to file

Reality Check: Which Diags Are Essential?

- linear runs: `Growth rate/Frequency`, `Amplitude spectra`, `Amplitude profiles`
- nonlinear runs: `Contour plots`, `Flux spectra`, `Flux profile`

Note: for eigenvalue analysis, enter `-[EV number]` as time window

Some basic checks & examples

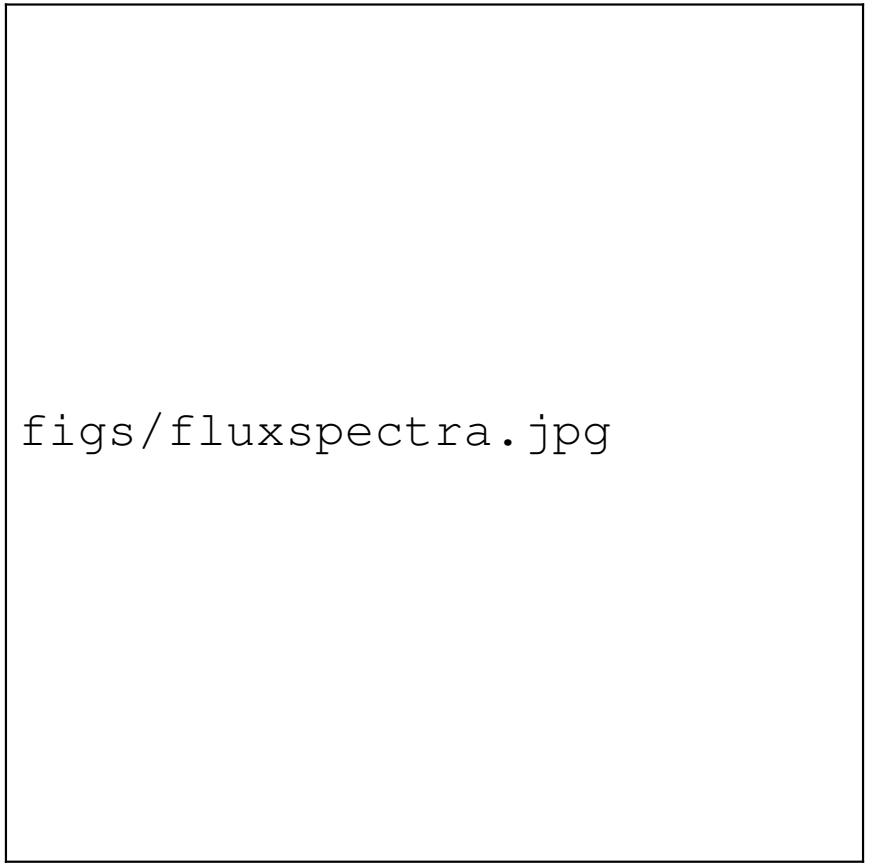
Check the metric coefficients for realistic geometries

figs/geometry.jpg

Linear: Check ballooning structure

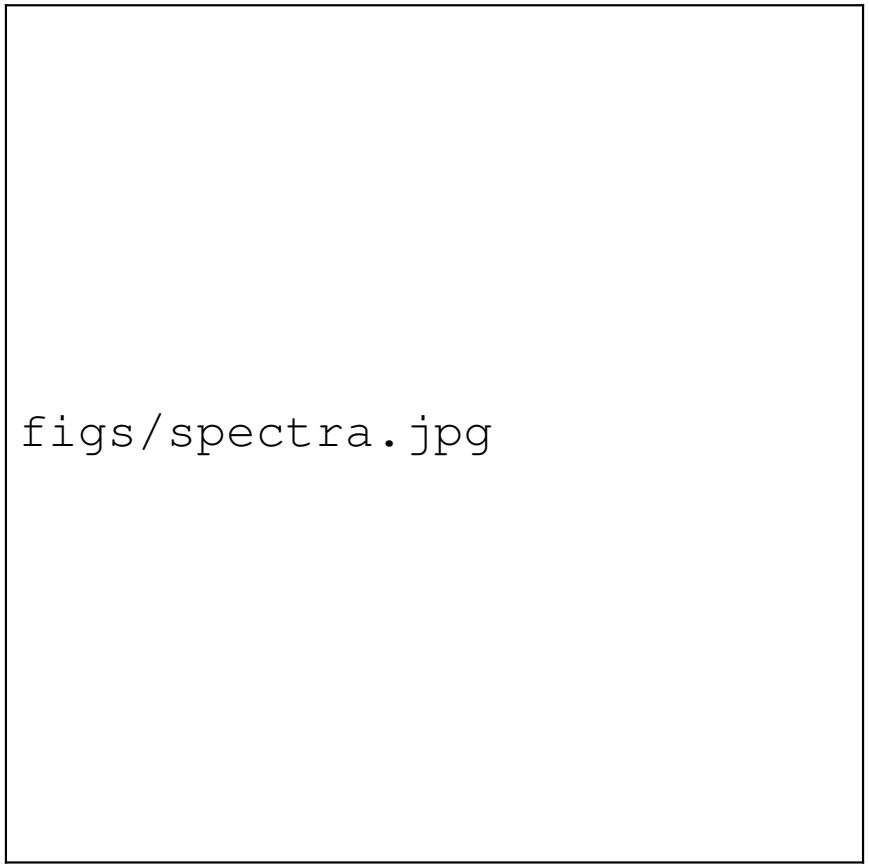
- select “Ballooning modes” for the last time steps
- mode structure should significantly fall off; if not increase `nx0`
- if still jagged afterwards, increase `nz0` or (carefully) apply `hyp_z`

Nonlinear: Check flux spectra



- select “Flux Spectra (ky/kx)”
- (!)use long time windows *beyond the overshoot* for nonlinear analysis(!)
- transport should be small at highest k_y, k_x, otherwise increase `nx0`, `nky0`
- transport should not peak at lowest k, increase box size `lx`/decrease `kymin` else

Nonlinear: Check amplitude spectra



- select “Amplitude spectra (ky/kx)”
- (!)use long time windows *beyond the overshoot* for nonlinear analysis(!)
- observables should be small at highest k_y, k_x and a power law should typically develop
- otherwise increase `nx0`, `nky0`

Happy computing!

Feel free to contact support@genecode.org if you have questions or comments

Appendix

The parallel boundary condition

Standard flux tube (flux bundle) parallel boundary condition

$$F(\rho, \nu, \chi + 2\pi) = F(\rho, \nu - 2\pi q, \chi)$$

with flux surface label ρ, field line label ν, and straight field line angle χ.

In GENE coordinates and Fourier transformed in 2nd dimension

$$F(x, k_y, z + L_z) = F(x, k_y, z) \exp\left(-2\pi \mathrm{i} n q(x)\right)$$

with toroidal mode number n, and radial/binormal/parallel coordinates $x/k_y/z$.

Local limit: Fourier transformed in 1st dimension and linearized safety factor

$F(k_x, k_y, z + L_z) = F(k'_x, k_y, z)(-1)^{\mathcal{N}j}$ with $k'_x = k_x + k_x^{\min}\mathcal{N}j$ and $\mathcal{N} = \hat{s}k_y^{\min}L_x$.

Here, $\hat{s}$ corresponds to `shat`, $\mathcal{N}$ to the optional input parameter `nexc` and j denotes the index in k_y.

Note that k_x wave numbers become coupled due to the parallel b.c.!
A mode might require several connections, see ballooning diagnostic!