Tricks used in GK simulations and implications for NIMROD

Scott Kruger

NIMROD
Madison, WI
June 2012

Motivation

- I’ve been using GYRO as part of another project, and there is one thing that sticks out:

Motivation

- I've been using GYRO as part of another project, and there is one thing that sticks out:

  Running GYRO is significantly easier than running NIMROD

Motivation

- I’ve been using GYRO as part of another project, and there is one thing that sticks out:

Running GYRO is significantly easier than running NIMROD

Why?

GK simulations of tokamak turbulence have relatively uniform solutions

1. Solutions of GK equations in tokamak core are very homogenous
   - No current sheets
   - ⇒ Gridding requirements are much more predictable (no packing, no worries about grid quality, …)
2. GYRO, GEM, … do not go beyond separatrix
   - Another considerable geometric simplification

GK simulations are very much like
simulating turbulent flow in a pipe

But there are other reasons why GYRO is easier than NIMROD

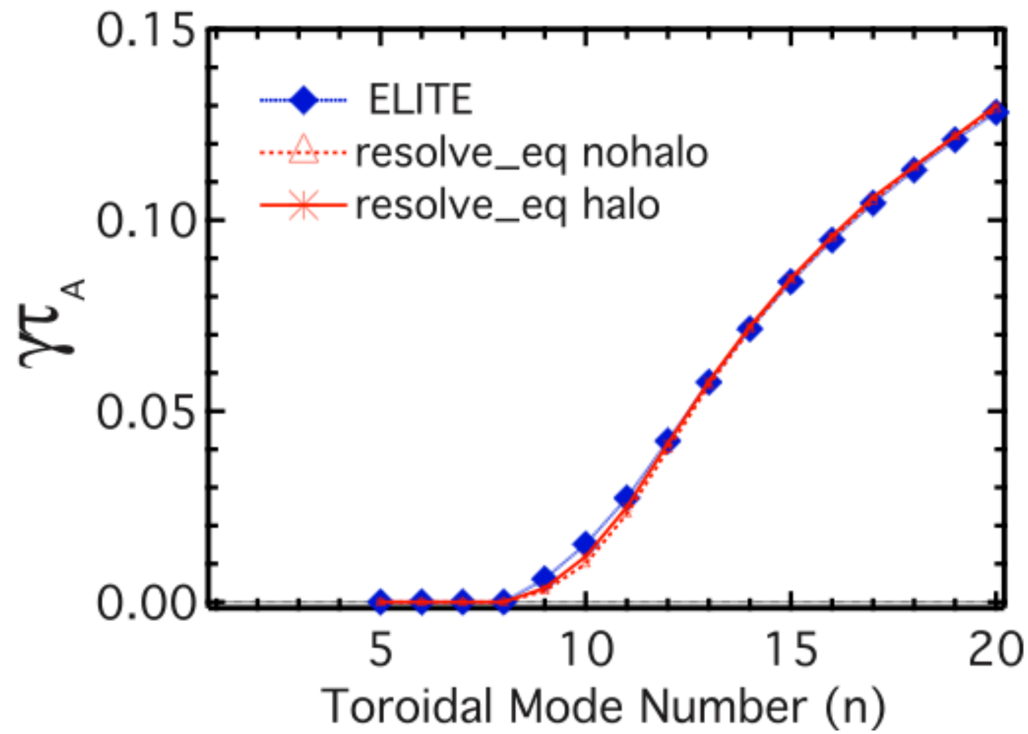
- Use of Miller equilibria
  - Two parts here:
    - Miller parameterization of a flux surface shape in terms of R_o, Z_o, κ, δ, ζ
    - Calculation of GS equilibria about a surface described by $R(\theta)$, $Z(\theta)$
- Use of eikonal representation
  - A.k.a. field-line following coordinates
  - Includes the use of simulating parts of torus (Δn)

Miller equilibria references

- Miller et.al. PoP **5**, 973 (1998)
  - Original paper motivated by Green-Chance analysis of ballooning modes
- Waltz and Miller, PoP **6**, 4265 (1999)
  - Specific discussion of how to use in GK simulations
- Hegna, PoP 7, 3921 (2000)
  - Generalization to 3D geometry
- Candy PPCF **51**, 105009 (2009).
  - Detailed discussion suitable for coding up a GK code

Key step: Taylor expand about the rational surface

- Miller local equilibria is a Taylor expansion of the Grad-Shafranov equilibria about the surface described by R(θ), Z(θ)
- Use of it for benchmark effectively is like recalculating equilibrium
  - No mapping errors to worry about
  - They didn't have to write their own equilibrium code



Paramaterization of $R(\theta)$,$Z(\theta)$ makes it even easier

- In practice, Miller equilibria is typically used in conjunction with parameterization of surfaces:
- $R(r,\theta)=R_o(r) + r\cos[\theta + \arcsin[\delta(r)]\sin(\theta)]$
  $Z(r,\theta)=Z_o(r) + \kappa\, r \sin[\theta + \zeta(r)\sin(2\theta)]$
- For theoretical studies, you only need to specify shaping parameters: R_o, Z_o, κ, δ, ζ
  - No TOQ, No EFIT, …

For many tokamak modes, eikenol representation is a great discretization method

- Contrast with NIMROD (neglecting the need for quantities to be real):

$$f = \sum \hat{f}(R,Z)e^{-in\phi} = \sum \hat{f}(r,\theta)e^{-in\phi}$$

- The eikonol expansion is

$$f = \sum \hat{f}_e(r,\theta)e^{-in\alpha}$$

Where $\alpha=\phi-\nu(r,\theta)$ and $\nu = F\int_0^\theta \frac{J}{R^2}d\theta'$

- Seems very similar, but the point is that functions in tokamaks oscillate along field lines
  - Number of grid point in θ is high for f, very low for f_e
  - ELITE for example requires fewer theta points than NIMROD
  - Similar in some ways to what the RF guys do (see Jenkins talk)

Note that boundary conditions are a bit more complicated for the envelope function

- α is not periodic
- Envelope function needs to have non-periodic boundary conditions
  - $f_e(r,\theta=0)=f_e(r,2\pi)\exp[-2\pi i n q(r)]$

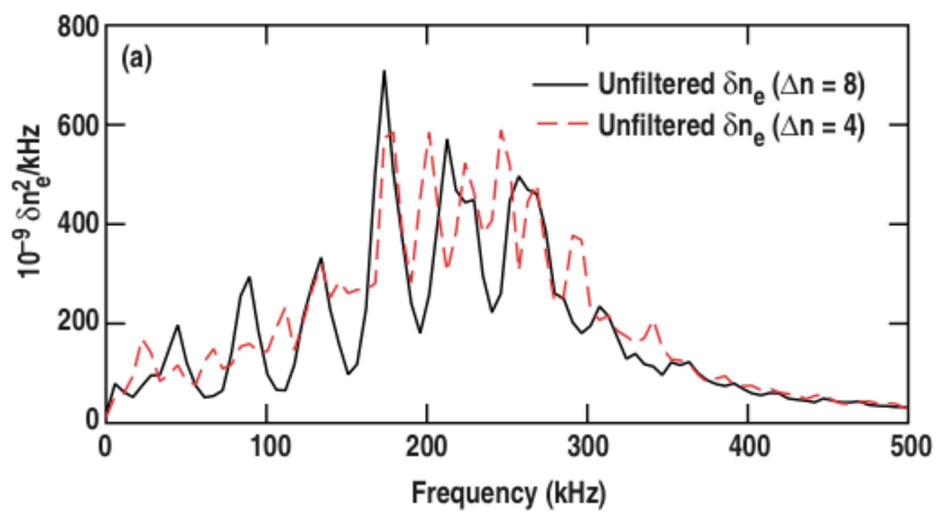
Eikonol representation usually combined with section of a torus

- Rather than do all the mode numbers, they usually do

$$f = \sum \hat{f}_e(r,\theta) e^{-i\frac{n}{\Delta n}\alpha}$$

- This enables solving for n=0,5,10,15,…
  - Hidden assumption is that when modes go unstable, there is only one dominant mode to worry about
  - Turbulence simulations converge nicely with this

From Holland 2009



What can NIMROD take away from this?

- Miller equilibria
  - It would be useful to start NIMROD some NIMROD simulations from parameterized Miller equilibra
    - PB simulations
    - Toroidal ITG
  - I'm exploring an easy way to do this with fluxgrid
    - Also see if we can improve the mapping quality
- Similarly, eikenol representation might be useful for the same simulations
  - Outline of plan:
    - keff(imode) -> keff(R,Z,imode)
    - Boundary condition routines modified