Modeling 2D Drift-Wave Turbulence with the Terry-Horton Equations

Catherine Felce (University of Oxford), advised by Prof. Greg Hammett & Noah Mandell (Princeton Plasma Physics Laboratory)

GOALS

- Simulate 2D drift-waves with a linear driving term.
- Develop model to describe turbulence on both ITG and ETG (Ion/Electron Temperature Gradient) scales without solving for all wavenumbers in between.

BACKGROUND:

2D Drift-Wave Model

- We use the Terry-Horton equations, derived as follows (1), (2), (3), (4).
- Conservation of ion-guiding-center density gives:
  \[
  \frac{\partial n_{gc}}{\partial t} + \mathbf{V} \cdot \left[ n_{gc} (\mathbf{V} \times \mathbf{B} + u|| \phi) \right] = 0
  \]
- Here \( u|| \) is the parallel ion flow, and ion-guiding-center density is related to the real-space ion density in the following way due to ion-polarization effects:
  \[
  \mathbf{P}_i = n_{gc} \mathbf{B} \times \mathbf{E}
  \]
- The drift velocity is given by:
  \[
  \mathbf{V} \times \mathbf{B} = \frac{c}{B} \mathbf{E} \times \nabla \phi
  \]
- Adding a linear drive, characterized by the delta_0 parameter, to the abadiadic (Boltzmann) electron distribution gives the following electron response:
  \[
  \delta n_e = n_{eo} \left( 1 - \delta \rho \frac{\partial}{\partial y} \right) \frac{c \phi}{T_e}
  \]
- Quasi-neutrality gives the following relation between ion density and potential:
  \[
  n_{gc} = n_{eo} \left( 1 - \rho \frac{\partial}{\partial y} \right) \frac{c \phi}{T_e}
  \]
- We approximate the ion-guiding-center density as a linear function of \( x \) (the radial direction), having a gradient scale length \( L_n \), and we consider the short-scale variations of ion density. We introduce the diamagnetic drift velocity \( V_d \), and describe \( u|| \) via a Landau damping model. Adding a dissipative term involving viscosity (\( \mu \)) we arrive at the following:
  \[
  \frac{\partial n_{gc}}{\partial t} + \mathbf{V} \times \mathbf{B} \cdot \nabla n_{gc} + n_0 V_d \frac{\partial}{\partial y} \left( \frac{c \phi}{T_e} \right) = -\alpha \frac{c_a}{L_n} n_{gc} + \mu \frac{c_s^2}{L_n} \nabla^2 n_{gc}
  \]

TRANSITION TO TURBULENCE

- From random initial conditions, during the stage of linear instability, density waves form in the poloidal (y) direction, appearing as thin, long ‘streamers’ (shown right (a)).
- These waves grow in amplitude by a factor of \( 10^4 \), according to the modes’ linear growth rates (\( \gamma \)). They then break into turbulent vortices (shown right (b)), due to a Kelvin-Helmholtz secondary instability. (5), (6)
- This is because the non-linear term in the equation only becomes significant as the density perturbations become large (the simulation shows the deviation of the density from the background value).
- The vortices are slightly radially elongated and cause particle transport in the radial (x) direction. This particle flux saturates in the nonlinear turbulence stage.

MODIFIED ADIABATIC ELECTRON RESPONSE

- Following Dorland [5] and St-Onge [4], we modify the quasi-neutrality equation for the ky=0 modes such that electrons no longer respond to the zonal (ky=0) component of the potential, which prevents unphysical radial electron transport. For these modes we replace the quasi-neutrality equation with the following:
  \[
  \frac{\partial n_{gc}}{\partial t} + \mathbf{V} \times \mathbf{B} \cdot \nabla n_{gc} + n_0 V_d \frac{\partial}{\partial y} \left( \frac{c \phi}{T_e} \right) = 0
  \]
  For ky = 0 modes.
- This causes enhanced zonal flows and zonal flux shearing, and decreases the level of transport.
- These zonal flows are ExB flows in the poloidal (y) direction. For the ky=0 modes, phi is constant in the y direction, leaving:
  \[
  \mathbf{V} \times \mathbf{B} = \frac{c \phi}{B} \frac{\partial \phi}{\partial x}
  \]
  For ky = 0 modes.
- A comparison of the modified (1) and unmodified (2) cases is shown below.
- Parameters: delta_0 (driving instability)=0.45, alpha (models Landau damping)=0.035, mu (viscosity)=0.0001

CONCLUSIONS

- The equations were implemented using a framework called Dedalus.
- Dedalus solves differential equations using spectral methods. It is open-source, written in Python and MPI-parallelized. [7]
- Equations can be symbolically entered, initial conditions are easily set and the time step is calculated automatically at each iteration of the solver.

FUTURE WORK

- It would be useful to replace the delta_0 parameter with a function giving a dispersion relation such that long and short scale turbulence is modeled.
- The model should give the desired behaviour at both ITG and ETG scales.
- This requires growth rates which peak at wavelengths comparable to the electron and ion gyro-radius respectively.

REFERENCES

7. http://dedalus-project.org

CONTACT

- Catherine Felce: catherine.felce@merton.ox.ac.uk
- Noah Mandell: nnmandel@gmail.com
- Greg Hammet: hammett@pppl.gov