Teaching Computational Plasma Physics

Richard E. Denton

Dartmouth College

Abstract

I developed a curriculum for computational plasma physics using one-dimensional computer codes. With the exception of a function for the plasma Z function, all of the codes were simple enough that they could be totally understood. Projects included programming a linear dispersion code, an MHD code in dipole coordinates and a linear dispersion code for Alfven waves in dipole coordinates, and modifying and running an electrostatic particle code and hybrid code. A simple convection code was used to teach flux limiters. Through hands-on projects using animations, students experienced the physics rather than just learning about it. My codes and notes are available on request at richard dot denton @ dartmouth dot edu.

Course Objectives

Students will:

1. develop a basic understanding of how plasma computations and simulations are performed and of fundamental issues in the interpretation of results derived from computer simulation, e.g., effects of the choice of boundary conditions, numerical approach, and physical model;

2. learn how physical and numerical approximations are used to solve computational plasma problems, e.g. when an electrostatic approximation can be used, magnetohydrodynamic vs. multifluid vs. hybrid vs. particle-in-cell approaches; learn how to employ simple and idealized boundary conditions to extract physically meaningful results for a given problem;

3. become proficient in using standard computational tools and techniques for attacking a variety of problems in plasma physics (binning data, modeling, solving simple dispersion equations), and gain exposure to advanced techniques;

4. learn plasma physics through seeing the phenomena in plots and animations.

Course Outline

1. Basic programming
2. Characteristics of simulations, static versus dynamic, fluid versus particle, dimensionality, relevant scale lengths and timescales
3. 1D finite difference magnetohydrodynamic (MHD) code, MHD equations and normalization, dispersion relation for waves, periodic and symmetry boundary conditions, energy conservation, and graphics

   Homework: Simulate various MHD waves, changing the perturbation and boundary conditions
4. Generalized curvilinear coordinates and the dipole magnetic field

Homework: Modify the MHD code to solve for Alfven waves in dipole geometry

5. Shooting code dispersion solver for MHD waves

    Homework: Modify dispersion solver to solve for Alfven waves in dipole geometry

6. 1D finite difference convection code including flux limiting, Von Neumann stability analysis

    Homework: Compare stability of finite difference code to analytical predictions; modify code for higher order differencing; simulate shock waves

7. 1D electrostatic particle code, normalization of equations, loading Maxwellian distribution, boundary conditions, and energy conservation

    Homework: Calculate temperature of particle distribution; examine the "thermal instability"; simulate the two stream instability and compare the growth rate to analytical predictions; simulate Landau damping and examine the behavior of resonant particles

8. Electrostatic dispersion code for waves

   Homework: Compare damping rate from dispersion code to that of the particle code for Landau damping

9. Hybrid code (particle ions and inertialess fluid electrons), normalization of equations, loading a gyrotropic particle distribution, "Boris particle mover", dispersion relation for waves, boundary conditions and energy conservation

   Homework: Correct an initialization error in the code, simulate purely left or right hand polarized waves, compare the frequency of ways to the dispersion relation, simulate the electromagnetic ion cyclotron instability comparing the complex frequency to that of the electromagnetic dispersion code WHAMP

10. Discussion of other types of simulations, spectral codes, full particle (particle in cell) and drift and gyrokinetic codes, Vlasov codes, delta f codes, ray tracing codes

Common themes running through course

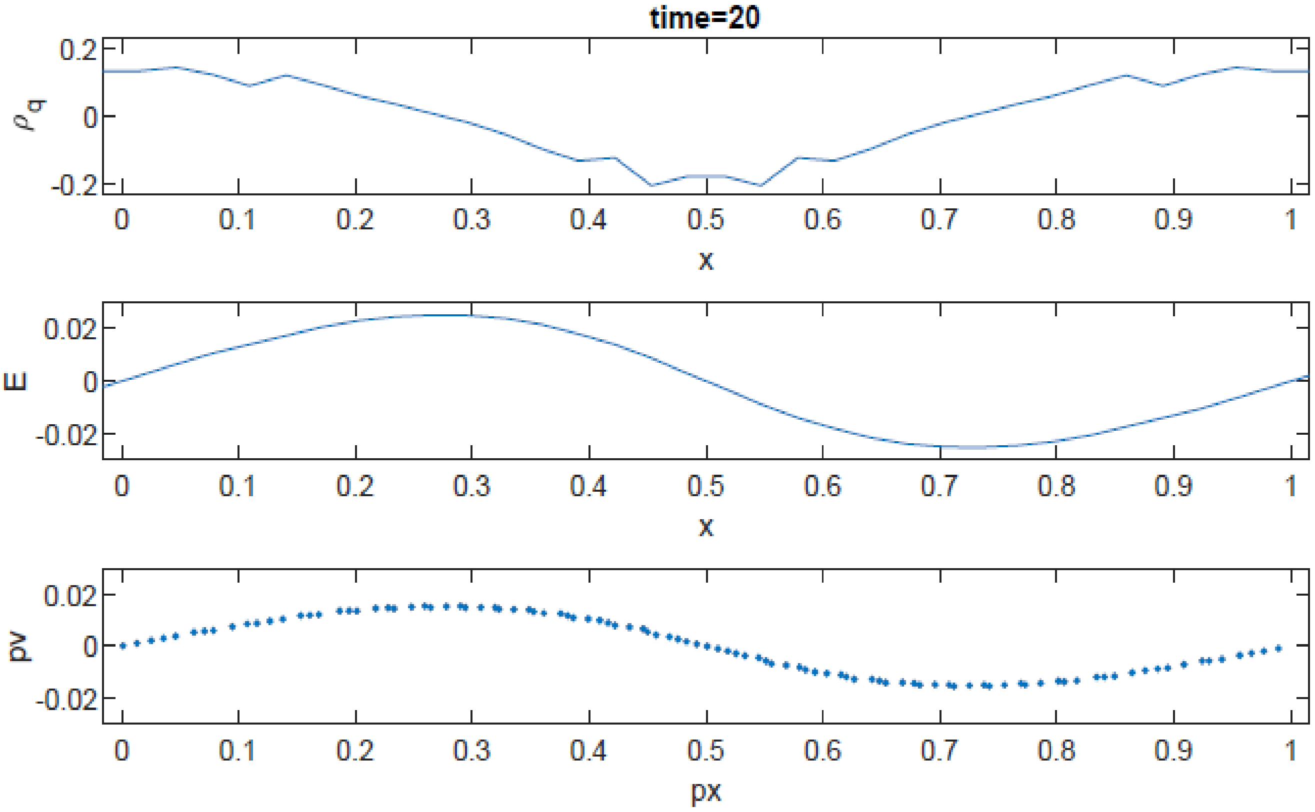
1. Different equations for different problems; validity of equations for different time and space scales
2. Normalization of equations
3. Effects of different boundary conditions
4. Energy conservation
5. Verifying results through comparisons with analytical theory or numerical dispersion codes
6. Representing results graphically

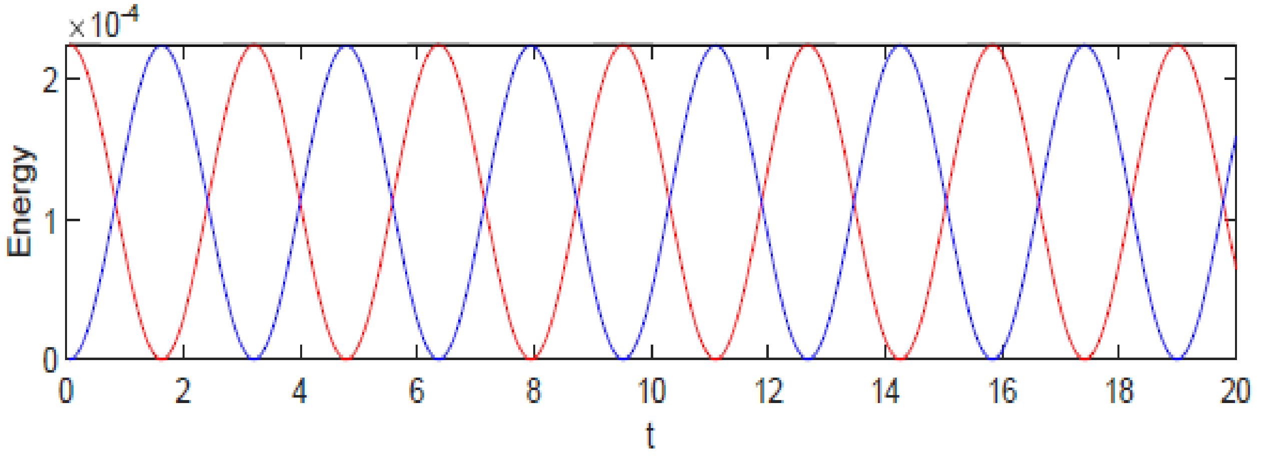
Example Plots from Electrostatic Particle Code

The students are given the code initialized to simulate a simple plasma wave. Below are the normalized charge density, electric field, and particle velocity for individual particles versus position, and the energy (red curve = kinetic energy, blue curve = electric field energy, black curve = total energy) versus time. When they run the code, they would see all of these plots changing with respect to time.

Then the students modify the code to include two populations of electrons streaming in opposite directions, to get the two stream instability.

Plots for Simple Plasma Wave





Plots for Two Stream Instability