Development of a GPU-Accelerated 3-D Full-Wave Code for Electromagnetic Wave Propagation in a Cold Plasma

D. Woodbury[1], S. Kubota[2], I. Johnson[3]

1. Department of Physics and Astronomy, Brigham Young University, Provo, UT 84604
2. Department of Physics and Astronomy, University of California, Los Angeles, Los Angeles, CA 90095
3. Princeton Plasma Physics Laboratory, Princeton University, Princeton, NJ 08540

Abstract

Computer simulations of electromagnetic wave propagation in magnetized plasmas are an important tool for both plasma heating and diagnostics. For active millimeter-wave and microwave diagnostics, accurately modeling the evolution of the beam parameters for launched, reflected or scattered waves in a toroidal plasma requires that calculations be done using the full 3-D geometry. Previously, we reported on the application of GPGPU (General-Purpose computing on Graphics Processing Units) to a 3-D vacuum Maxwell code using the FDTD (Finite-Difference Time-Domain) method [1]. In the current study, we have modified the 3-D code to an induced current density based on the cold plasma dispersion relation. Results from Gaussian beam propagation in an inhomogeneous anisotropic plasma are presented and compared to well known solutions. Additional enhancements and further research is also discussed.

Introduction

- Reflectometry measures plasma profiles and turbulence
- Microwave beams are directed into the plasma and reflected at a specific density layer (See Figure 1)
- Density is reconstructed from the phase and amplitude
- Synthetic diagnostics compare data to an accurate model
- The synthetic diagnostic must be run for hundreds of possible turbulence profiles
- Running the code on a GPU offers considerable speedup compared to serial implementations. A 3-D code offers more accuracy than simpler models

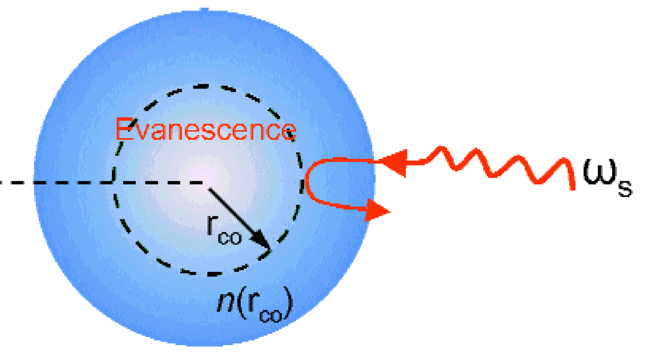

Figure 1: A beam of frequency ω will propagate in plasma until it reaches a cutoff density
http://tempest.das.ucdavis.edu/pdg/MIR/Images/plasma_reflect.gif

Equations and Methods

By normalizing all variables, we can write the equations for waves propagating in plasma under the cold plasma approximation as follows. These can be written to express equations for each individual field component

$$\frac{\partial \hat{\mathbf{B}}}{\partial \hat{t}} = -\hat{\nabla} \times \hat{\mathbf{E}} \qquad \frac{\partial \hat{\mathbf{E}}}{\partial \hat{t}} = \hat{\nabla} \times \hat{\mathbf{B}} - \hat{\mathbf{J}}$$

$$\frac{\partial \hat{\mathbf{J}}}{\partial \hat{t}} = -\hat{\nu}\hat{\mathbf{J}} - \alpha\, \hat{\mathbf{J}} \times \mathbf{b} + f\, \hat{\mathbf{E}}.$$

- A Yee lattice staggers the position of the field and current densities in both space and time
- Finite difference approximations of derivatives are used to obtain updates to the field
- For current densities, we start with the J-E Convolution approximation and rearrange the equations to express update equations. An example update is shown:

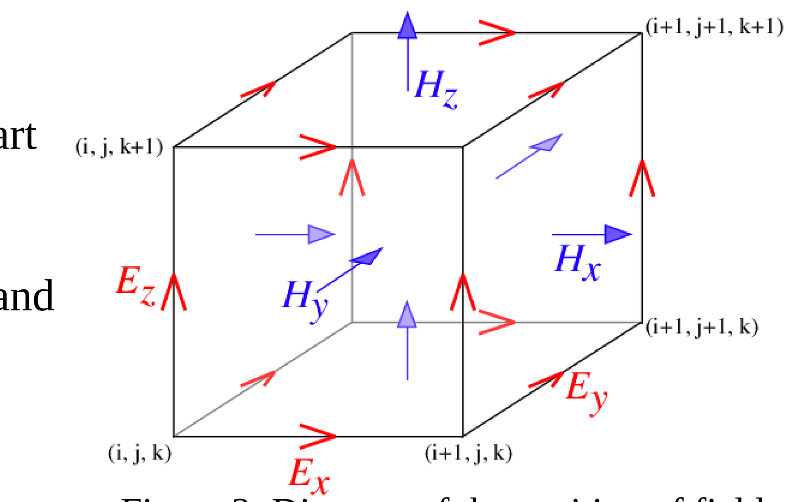

Figure 2: Diagram of the position of field vectors on a Yee lattice. http://ab-initio.mit.edu/wiki/images/6/6f/Yee-cube.png

$$J_x^{n+0.5} = \frac{(f\Delta t)e^{-\nu\Delta t/2}(\gamma_x^2+1)}{1+\gamma^2}E_x^n + \frac{(f\Delta t)e^{-\nu\Delta t/2}(\gamma_x\gamma_y-\gamma_z)}{1+\gamma^2}E_y^n + \frac{(f\Delta t)e^{-\nu\Delta t/2}(\gamma_z\gamma_x+\gamma_y)}{1+\gamma^2}E_z^n + \left[\frac{(1+e^{-\nu\Delta t})(\gamma_x^2+1)}{1+\gamma^2}-1\right]J_x^{n-0.5} + \frac{(1+e^{-\nu\Delta t})(\gamma_x\gamma_y-\gamma_z)}{1+\gamma^2}J_y^{n-0.5} + \frac{(1+e^{-\nu\Delta t})(\gamma_z\gamma_x+\gamma_y)}{1+\gamma^2}J_z^{n-0.5}$$

Code Architecture

- The code is written with NVIDIA's CUDA extension to C, which executes functions on a GPU card
- Function calls are sent to a block of processor cores on the GPU, providing general instructions for an update
- Each processor is directed to a specific memory location and updates the fields at a point in parallel with others
- The process is repeated iteratively until all of the data is again copied on to the CPU and can be analyzed

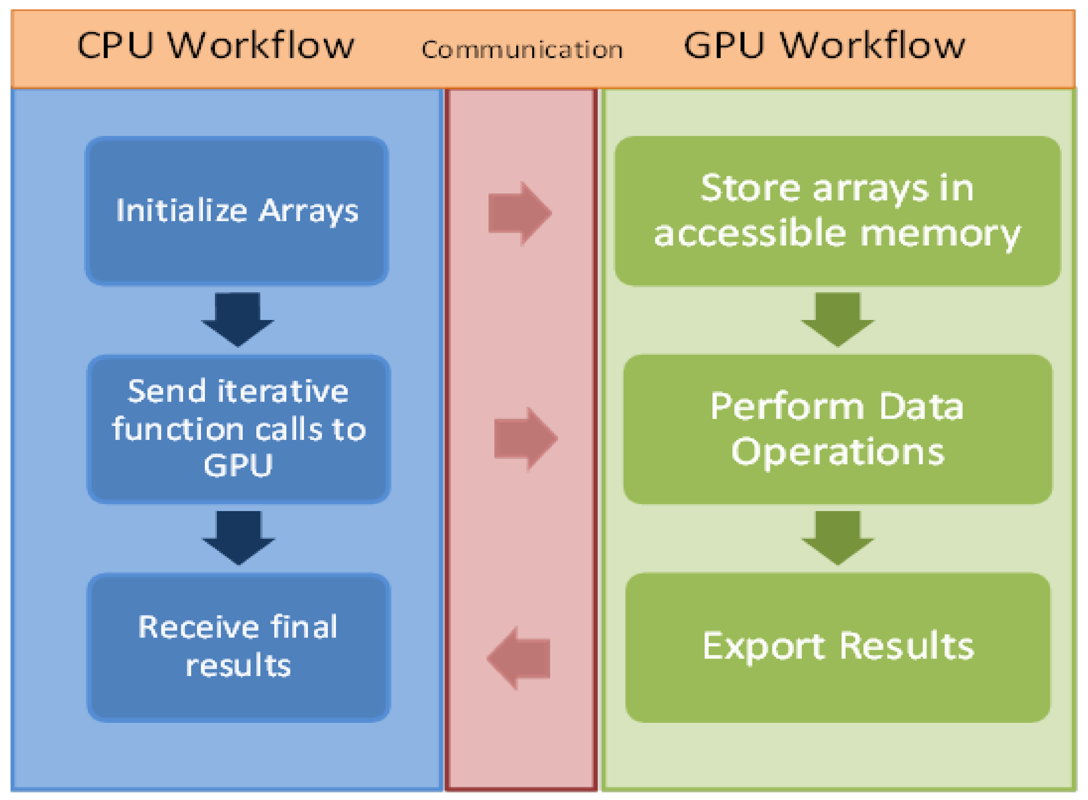

Figure 3: Code structure. Since memory transfer is computationally expensive, no data is transferred back until all iterations are complete

Theoretical Comparison

- We used plasma targets for which analytic or well approximated results are known.
- A wide beam waist produced plane wave like solutions
- Simulations were run on a NVIDIA K20m GPU. Grid size was 140x401x401 with 20 points per wavelength
- Wave frequency f was set to 40 GHz
- Data is shown below as surface plots and as colored squares when compared with solutions

Ordinary Mode Cutoff

- For waves polarized parallel to the magnetic field (O Mode), the wave will reflect completely when its frequency matches the plasma frequency
- For linearly varying plasma density, the steady state solution is an Airy function, as shown by Swanson [2]

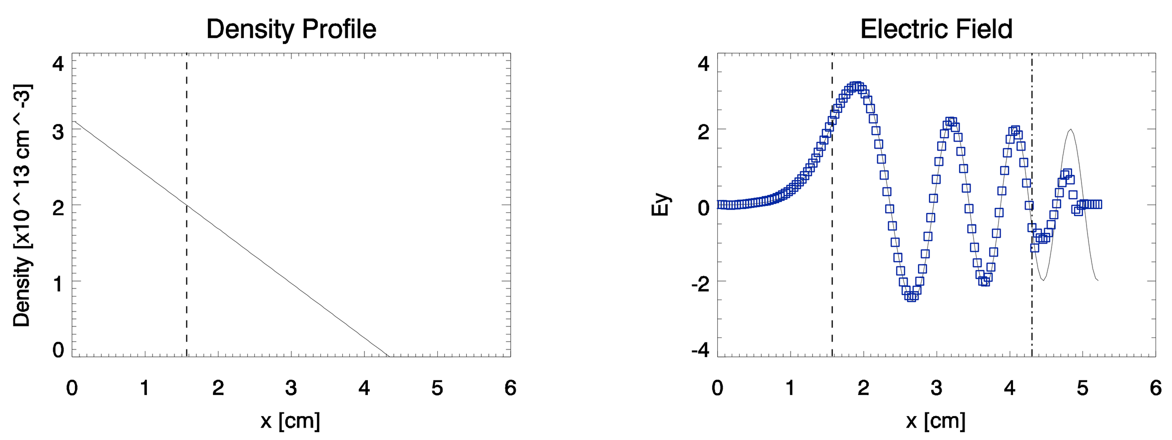

Figure 4. The cutoff density is indicated by the dotted line. The data is compared to a 1-D Runge-Kutte solver. Right of the dot-dash line (source grid), only reflected fields are calculated

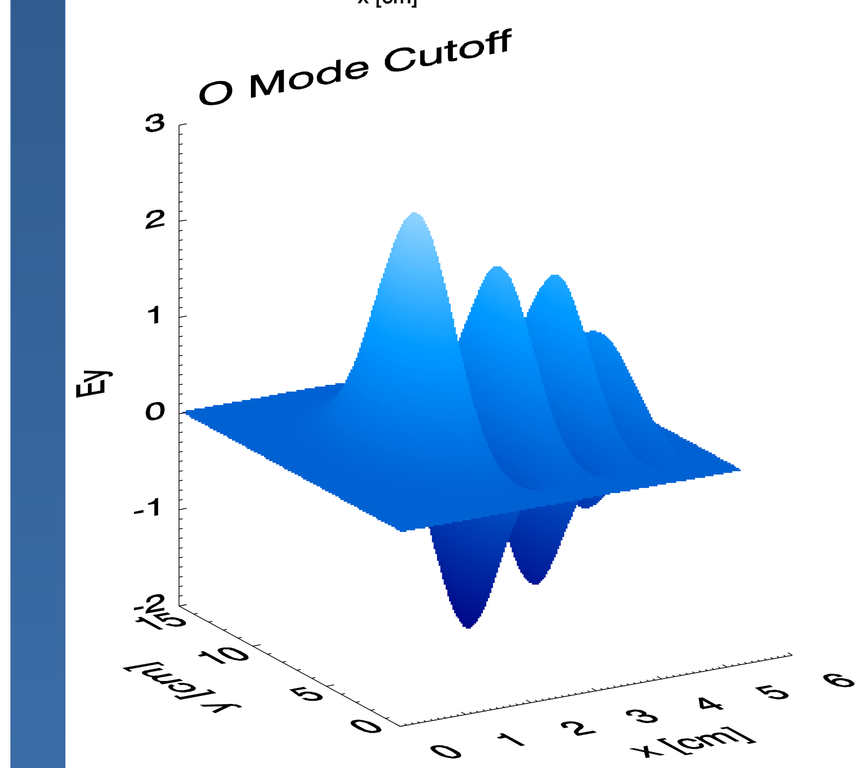

Figure 5. Electric field at plane through the beam center. The wavelength is visibly longer as the plasma density increases

Extraordinary Mode Cutoff

- For waves polarized perpendicular to the magnetic field (X mode), the wave excites $E_x \parallel$ **k** and the cutoff density is altered by the background magnetic field
- The cutoff frequency can be found using the following equation [2]

$$\omega_X = \left[\left(\frac{\omega_{ce}}{2}\right)^2 + \omega_p^2\right] \pm \frac{\omega_{ce}}{2}$$

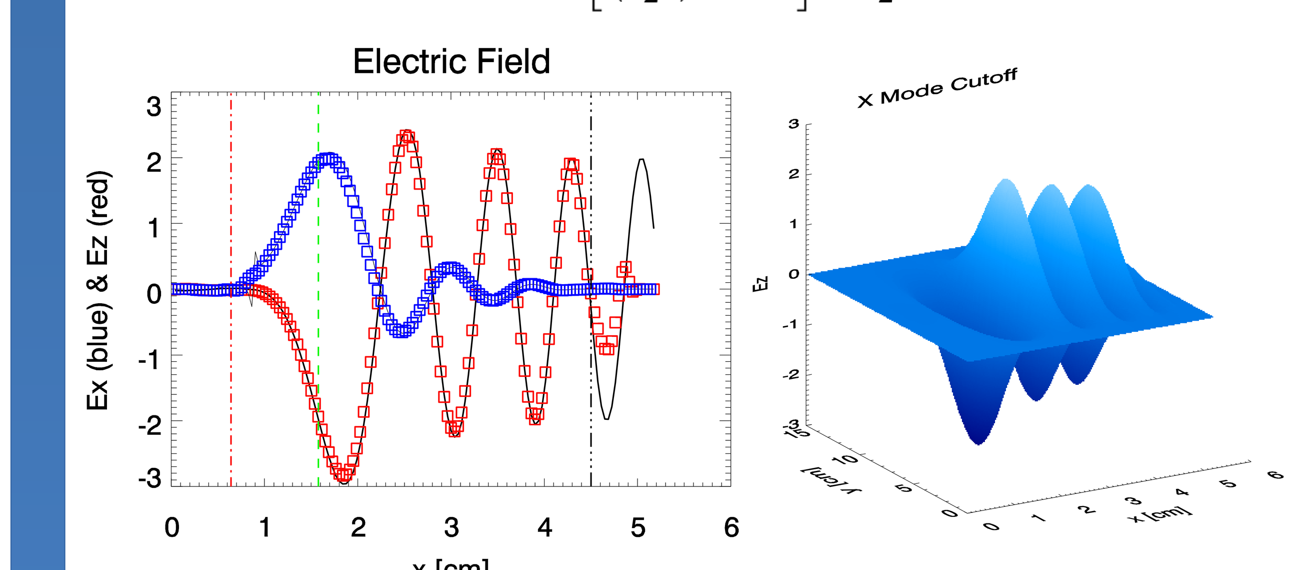

Figure 6. Data compared to the 1-D Runge-Kutte solver. The ordinary cutoff density is in red, with the new cutoff in green

Faraday Rotation

- For a plasma whose background magnetic field is aligned with **k**, the polarization will rotate due to a difference between right and left circular polarizations
- Equations for the index for the two modes and the change in polarization [2]

$$n_{\mathrm{R,L}}^2 = 1 - \frac{\omega_{\mathrm{pe}}^2}{\omega\,(\omega \mp \omega_{\mathrm{ce}})}$$

$$\frac{\mathrm{d}\phi}{\mathrm{d}z} = \frac{\mathrm{d}}{\mathrm{d}z}\left(\frac{\Delta n}{2}\frac{\omega}{c}z\right) = \frac{\omega}{2c}(n_L - n_R)$$

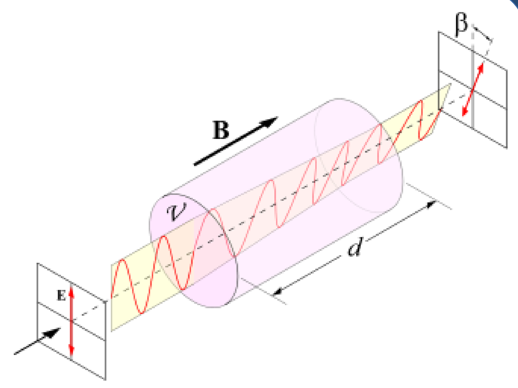

Figure 7. A schematic of the Faraday effect
http://upload.wikimedia.org/wikipedia/commons/c/cb/Faraday-effect.svg

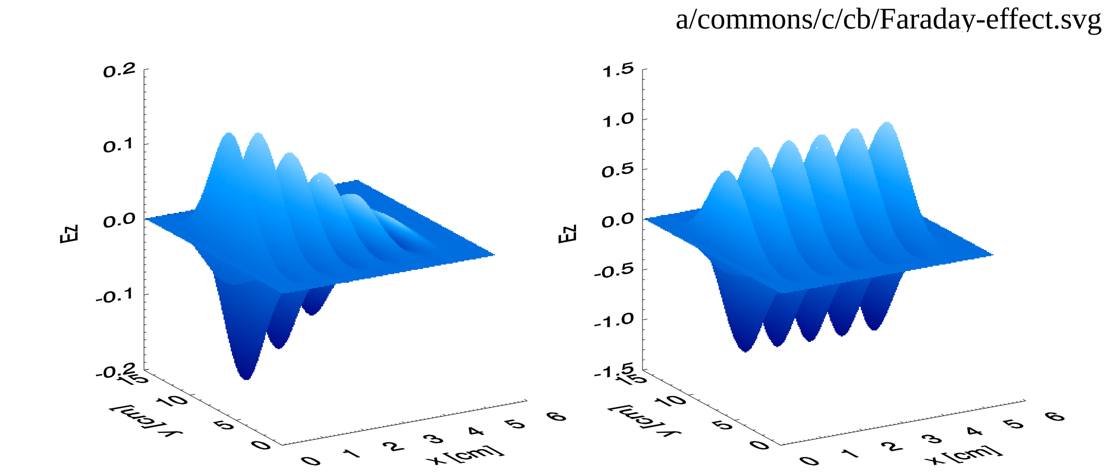

Figure 8. E_z increases from right to left as the wave travels

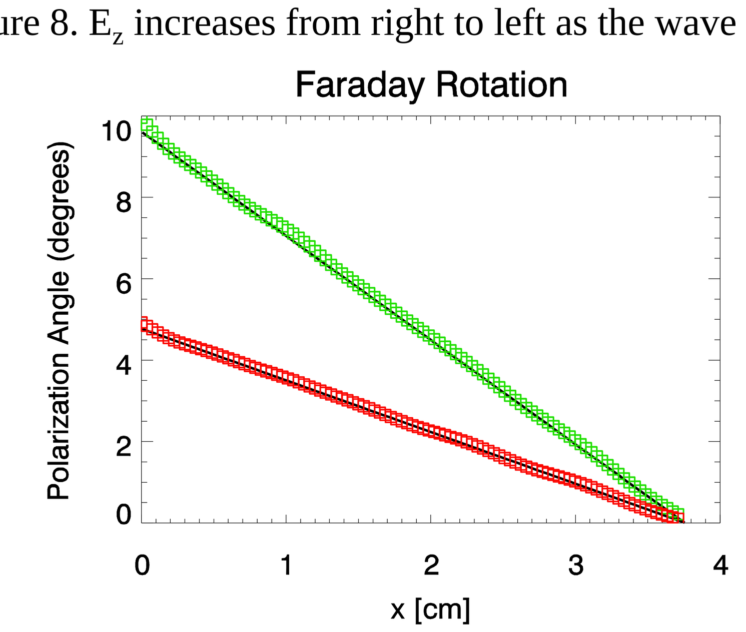

Figure 9. Polarization change for two simulation runs compared to theoretical values

Conclusion

- The code accurately models test cases for microwave diagnostics and can be applied to more complex geometries
- Possible enhancements include launching the beam at arbitrary angles, improving methods for loading plasma targets and profiling the code for additional speedup
- The application of the diagnostic to tokamak plasmas will require windowing or cluster techniques to avoid exceeding the onboard memory limits of GPU cards

This work supported by U.S. DoE Grant DE-FG02-99-ER54527 and in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship (SULI) program

References

[1] K.S. Reuther, et al., *Development of a GPU-Accelerated 3D Full-Wave Code for Reflectometry Simulations.* Poster JP8.00021, Presented at the 54th Annual Meeting of the APS Division of Plasma Physics, Nov. 11-15, 2013, Denver, CO
[2] D. G. Swanson, *Plasma Waves*, 2nd ed. (Institute of Physics, Bristol 2003)