Term project description

O. Koshkarov, I. Romadanov

University of Saskatchewan

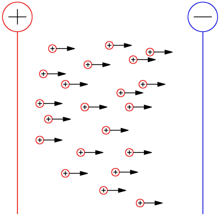
March 18, 2014

Abstract

The main goal of this project is analytical and numerical studies of instabilities of counter streaming plasma flows for emissive probe experiments. Hydrodinamic approximation is used to describe those systems. We use dimensionless form of equations for numerical simulation. To implement iterative numerical method the system of equations were split into boundary value problem and initial value problem. Multiple shooting method and upwind scheme are used respectively. High computational load forced us to parallelize our problem. MPI library was suggested to implement, because high efficiency of distributed memory system for such types of problems. Alternatively, shared memory and GPU systems are going to be considered as tools to simulate our problem.

Introduction

In the system of electrical diode, which consist of two charged electrodes with an ion beam between them,waves can arise. The main goal of this project is to numerically simulate such a system and investigate wave propagation. Some asymptotic frequencies and waves growth rates, from analytical approach could be obtained. It is interesting to compare these asymptotics with numerical results and obtain solutions for more difficult systems.To describe our system we will use the continuity and Euler equation for ion component of plasma, Poisson equation for electrostatic interaction, and Boltzmann equation for electron density.



System of equations mentioned above will be linearized and written for ion density n, ion velocity v and electrostatic potential perturbations ϕ:

$$\frac{\partial n}{\partial t} + v_0 \frac{\partial n}{\partial z} + n_0 \frac{\partial v}{\partial z} = 0 \tag{1}$$

$$\frac{\partial v}{\partial t} + v_0 \frac{\partial v}{\partial z} + \frac{e}{M} \frac{\partial \phi}{\partial z} = 0 \tag{2}$$

$$\frac{\partial^2 \phi}{\partial z^2} = -4\pi e \left(n - \frac{e n_0}{T} \phi \right) \tag{3}$$

with boundary conditions (z=0 or z=L for electrode location):

$$\phi(0) = \phi(L) = n(0) = v(0) = 0 \tag{4}$$

where n, v, ϕ - ion density, ion velocity and electrostatic potential perturbation, while n_0, v_0, are equilibrium ion density and velocity, and e, M, T are absolute values of electron charge, ion mass and electron temperature respectively.

For numerical study one should rewrite equation into dimensionless form. Let:

$$n^* = \frac{n}{n_0}; z^* = \frac{z}{\lambda_D}; \phi^* = \frac{e\phi}{T}; t^* = t\omega_p \tag{5}$$

where $\omega_p^2 = \frac{4\pi e^2 n_0}{M}$, $\lambda_D^2 = \frac{T}{4\pi e^2 n_0}$ This transform our system into (stars were dumped for convenience):

$$\frac{\partial n}{\partial t} + v_0 \frac{\partial n}{\partial z} + \frac{\partial v}{\partial z} = 0 \tag{6}$$

$$\frac{\partial v}{\partial t} + v_0 \frac{\partial v}{\partial z} + \frac{\partial \phi}{\partial z} = 0 \tag{7}$$

$$\frac{\partial^2 \phi}{\partial z^2} = \phi - n \tag{8}$$

Above system could be separated on:

(a) Boundary value problem:

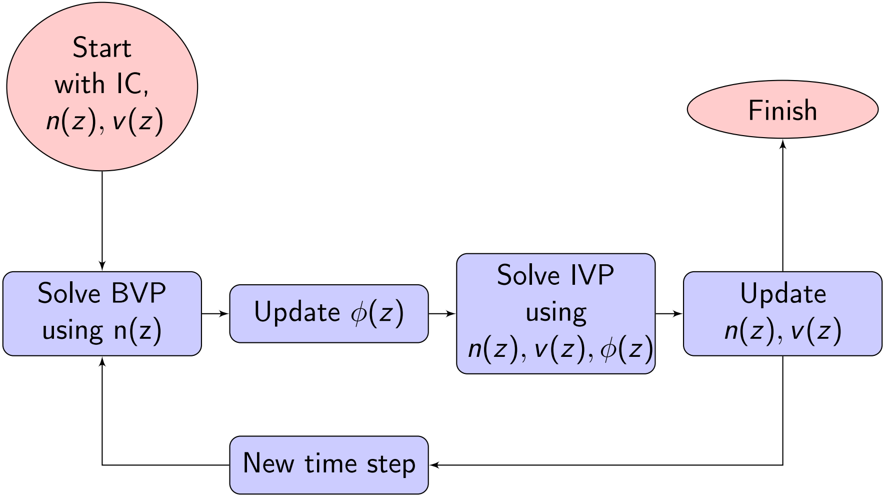
$$\begin{cases} \frac{\partial^2 \phi}{\partial z^2} = \phi - n \\ \phi(0) = \phi(L) = 0 \\ n(z) & \text{-given} \end{cases} \tag{9}$$

(b) Initial value problem:

$$\begin{cases} \frac{\partial n}{\partial t} + v_0 \frac{\partial n}{\partial z} + \frac{\partial v}{\partial z} = 0 \\ \frac{\partial v}{\partial t} + v_0 \frac{\partial v}{\partial z} + \frac{\partial \phi}{\partial z} = 0 \\ \phi(z), n(t=0,z), v(t=0,z), \quad \text{-given} \end{cases} \tag{10}$$

Some special initial conditions, which are obtained from theory could be used to trigger instability.

The solution could be represented as following:



To solve (9) numerically, it is easier to rewrite it in the form:

$$\begin{cases} \frac{\partial \mathbf{q}}{\partial z} = A \cdot \mathbf{q} + \mathbf{b} \\ B_0 \cdot \mathbf{q}(0) + B_L \cdot \mathbf{q}(L) = 0 \\ \mathbf{b}(z) & -\text{given} \end{cases} \tag{11}$$

where

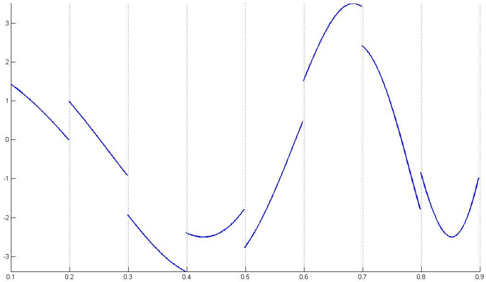
$$\mathbf{q} = \begin{pmatrix} \phi \\ \frac{\partial \phi}{\partial z} \end{pmatrix}, \ \mathbf{b} = -\begin{pmatrix} 0 \\ n \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ B_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \ B_L = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

To solve BVP (Boundary value problem) multiple shooting method was used.

Numerical methods

Multiple shooting method

To apply multiple shooting one should divide the domain into many small subintervals. On every interval we have IVP with artificial initial conditions. By imposing equations, which provide continuity of different part of solutions and satisfying of boundary conditions, we obtain number of algebraic equation to solve. We solve original BVP by solving this system (for example using Newton's method)



To solve (10) numerically let rewtite it in the form:

$$\begin{cases} \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial z} = 0 \\ \mathbf{U}(0, z), \mathbf{F} & -\text{given} \end{cases} \tag{12}$$

where

$$\mathbf{U} = \begin{pmatrix} n \\ v \end{pmatrix}, \mathbf{F} = \begin{pmatrix} v_0 n + v \\ v_0 v + \phi \end{pmatrix}.$$

To solve IVP (Initial value problem) upwide scheme was used:

$$\mathbf{U}_j^{n+1} = \mathbf{U}_j^n + \frac{\Delta t}{\Delta z}(\mathbf{F}_j^n - \mathbf{F}_{j-1}^n) \tag{13}$$

Parallel implementation

Parallelization of numerical methods using MPI.

(a) The idea of MPI parallelization for multiple shooting:

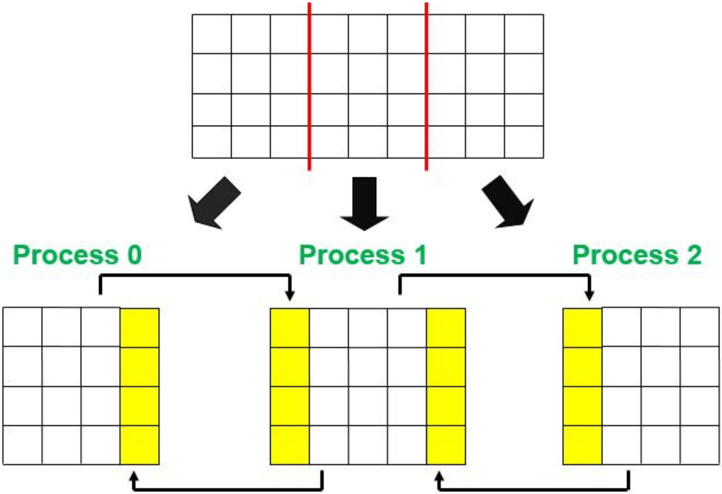
1. Solve IVP on each subinterval on different processor.
2. To solve system of algebraic equations we can parallelize a Gaussian elimination.

(b) The idea of MPI parallelization for upwind scheme:

1. We should divide our space domain into equal pieces for each for processor.
2. Scatter the initial conditions on those subdomains with some boundary buffer.
3. Solve all subdomains separately
4. Gather all necessary information
5. Repeat

Parallel implementation

MPI parallelization for upwind scheme



Results

Video for Gaussian IC

Results

Video for uniform distributed IC

Future work

1. **Future work**
   (a) Parallelization of numerical methods using OpenMP.
   (b) Computing on GPU using CUDA.
2. **Result verification**
   (a) Comparing with analytical results
   (b) Comparing with results obtained from BOUT.