Monte Carlo Simulation of the Two Stream Instability

Libor Novák
Advisor: Prof. RNDr. Petr Kulhánek, CSc

Research project

Building a Computer Model

To get solution of a physical problem using computer simulation requires performing this process:

Building a Computer Model

To get solution of a physical problem using computer simulation requires performing this process:

- **Problem formulation**

Building a Computer Model

To get solution of a physical problem using computer simulation requires performing this process:

- **Problem formulation**
- **Building a model**

Building a Computer Model

To get solution of a physical problem using computer simulation requires performing this process:

- **Problem formulation**
- **Building a model**
- **Solving the model**

Building a Computer Model

To get solution of a physical problem using computer simulation requires performing this process:

- **Problem formulation**
- **Building a model**
- **Solving the model**
- **Comparison of the model and experimental or theoretical results**

Computer Simulation of Plasmas

This branch of study is based on 2 different approaches and their combination:

Computer Simulation of Plasmas

This branch of study is based on 2 different approaches and their combination:

- **Fluid simulation** - numerical solving of the magnetohydrodynamic equations

Computer Simulation of Plasmas

This branch of study is based on 2 different approaches and their combination:

- **Fluid simulation** - numerical solving of the magnetohydrodynamic equations
- **Kinetic simulation** - numerical solving of the plasma kinetic equations (Vlasov, Fokker - Planck, ...) or particle simulation

Computer Simulation of Plasmas

This branch of study is based on 2 different approaches and their combination:

- **Fluid simulation** - numerical solving of the magnetohydrodynamic equations
- **Kinetic simulation** - numerical solving of the plasma kinetic equations (Vlasov, Fokker - Planck, ...) or particle simulation
- **Hybrid simulation** - combination of both approaches

Particle Simulations

Divided into 2 areas and their combination:

Particle Simulations

Divided into 2 areas and their combination:

- **Molecular dynamics method** - numerical solving of charged particles motions, interacting with each other and with externally fields

Particle Simulations

Divided into 2 areas and their combination:

- **Molecular dynamics method** - numerical solving of charged particles motions, interacting with each other and with externally fields
- **Monte Carlo method** - probability description by a random quantity

Particle Simulations

Divided into 2 areas and their combination:

- **Molecular dynamics method** - numerical solving of charged particles motions, interacting with each other and with externally fields
- **Monte Carlo method** - probability description by a random quantity
- **Hybrid simulation** - combination of both methods

Monte Carlo Method

Solving a problem by Monte carlo method consists from these steps:

Monte Carlo Method

Solving a problem by Monte carlo method consists from these steps:

- **Problem analysis and model creation**

Monte Carlo Method

Solving a problem by Monte carlo method consists from these steps:

- **Problem analysis and model creation**
- **Random quantity generation**

Monte Carlo Method

Solving a problem by Monte carlo method consists from these steps:

- **Problem analysis and model creation**
- **Random quantity generation**
- **Random quantity transformation**

Monte Carlo Method

Solving a problem by Monte carlo method consists from these steps:

- **Problem analysis and model creation**
- **Random quantity generation**
- **Random quantity transformation**
- **Previous 2 steps repeating and statistical evaluation of the results**

Random Numbers Generators

Random numbers can be generated through several ways. In Monte carlo methods were or are used:

Random Numbers Generators

Random numbers can be generated through several ways. In Monte carlo methods were or are used:

- **Physical generators**

Random Numbers Generators

Random numbers can be generated through several ways. In Monte carlo methods were or are used:

- **Physical generators**
- **Random numbers tables**

Random Numbers Generators

Random numbers can be generated through several ways. In Monte carlo methods were or are used:

- **Physical generators**
- **Random numbers tables**
- **Calculated random numbers**

Pseudo-random Numbers Generators 1

Good generator should have these features:

Pseudo-random Numbers Generators 1

Good generator should have these features:

- **Long period**

Pseudo-random Numbers Generators 1

Good generator should have these features:

- **Long period**
- **Uniform distribution of the numbers**

Pseudo-random Numbers Generators 1

Good generator should have these features:

- **Long period**
- **Uniform distribution of the numbers**
- **Numbers are not correlated**

Pseudo-random Numbers Generators 1

Good generator should have these features:

- **Long period**
- **Uniform distribution of the numbers**
- **Numbers are not correlated**
- **High speed of evaluation**

Pseudo-random Numbers Generators 2

Pseudo-random Numbers Generators 2

General Formula

$$n_i = f(n_{i-1}, n_{i-2}, ..., n_{i-j}) \tag{1}$$

Pseudo-random Numbers Generators 2

General Formula

$$n_i = f(n_{i-1}, n_{i-2}, ..., n_{i-j}) \tag{1}$$

Linear Congruential Generators (LCG)

$$n_i = (an_{i-1} + b)\mathrm{mod}(m) \tag{2}$$

Pseudo-random Numbers Generators 2

General Formula

$$n_i = f(n_{i-1}, n_{i-2}, ..., n_{i-j}) \tag{1}$$

Linear Congruential Generators (LCG)

$$n_i = (an_{i-1} + b)\mathrm{mod}(m) \tag{2}$$

Lagged Fibonacci Generators (LFG)

$$n_i = (an_{i-1} + bn_{i-2} + ...)\mathrm{mod}(m) \tag{3}$$

Starting the Binary Collision

Starting the Binary Collision

The Relative Velocity

$$\boldsymbol{u}^t = \boldsymbol{v}^t_\alpha - \boldsymbol{v}^t_\beta \tag{4}$$

Starting the Binary Collision

The Relative Velocity

$$\boldsymbol{u}^t = \boldsymbol{v}^t_\alpha - \boldsymbol{v}^t_\beta \tag{4}$$

Rotation

$$\begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^t = \begin{pmatrix} 0 \\ 0 \\ u \end{pmatrix}^t \tag{5}$$

Starting the Binary Collision

The Relative Velocity

$$\boldsymbol{u}^t = \boldsymbol{v}_\alpha^t - \boldsymbol{v}_\beta^t \tag{4}$$

Rotation

$$\begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^t = \begin{pmatrix} 0 \\ 0 \\ u \end{pmatrix}^t \tag{5}$$

Inverse Matrix

$$\boldsymbol{I} = \begin{pmatrix} \cos\theta\cos\varphi & -\sin\varphi & \sin\theta\cos\varphi \\ \cos\theta\sin\varphi & \cos\varphi & \sin\theta\sin\varphi \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{6}$$

Random Angles Generation

$(0, 0, u)^t$ Rotation by Φ and Θ

$$\sin\Theta = \frac{2\delta}{1+\delta^2} \qquad (7)$$

$$1 - \cos\Theta = \frac{2\delta^2}{1+\delta^2} \qquad (8)$$

Random Angles Generation

$(0, 0, u)^t$ Rotation by Φ and Θ

$$\sin\Theta = \frac{2\delta}{1+\delta^2} \tag{7}$$

$$1-\cos\Theta = \frac{2\delta^2}{1+\delta^2} \tag{8}$$

$$\delta = \tan\Theta/2 \tag{9}$$

$$\sigma^2 = \frac{e_\alpha^2 e_\beta^2 n_L \lambda}{8\pi\epsilon_0^2 m_{\alpha\beta}^2 u^3}\Delta t \tag{10}$$

Results of the Binary Collision

$$\begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^{t+\Delta t} = \boldsymbol{I} \begin{pmatrix} u \sin \Theta \cos \Phi \\ u \sin \Theta \sin \Phi \\ u \cos \Theta \end{pmatrix} \tag{11}$$

Results of the Binary Collision

$$\begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^{t+\Delta t} = \boldsymbol{I} \begin{pmatrix} u \sin\Theta \cos\Phi \\ u \sin\Theta \sin\Phi \\ u \cos\Theta \end{pmatrix} \tag{11}$$

$$\begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^{t+\Delta t} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^{t} + \begin{pmatrix} \Delta u_x \\ \Delta u_y \\ \Delta u_z \end{pmatrix} \tag{12}$$

Results of the Binary Collision

$$\begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^{t+\Delta t} = \boldsymbol{I} \begin{pmatrix} u \sin\Theta \cos\Phi \\ u \sin\Theta \sin\Phi \\ u \cos\Theta \end{pmatrix} \tag{11}$$

$$\begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^{t+\Delta t} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}^{t} + \begin{pmatrix} \Delta u_x \\ \Delta u_y \\ \Delta u_z \end{pmatrix} \tag{12}$$

$$\boldsymbol{v}_\alpha^{t+\Delta t} = \boldsymbol{v}_\alpha^t + \frac{m_{\alpha\beta}}{m_\alpha}\Delta\boldsymbol{u} \tag{13}$$

$$\boldsymbol{v}_\beta^{t+\Delta t} = \boldsymbol{v}_\beta^t - \frac{m_{\alpha\beta}}{m_\beta}\Delta\boldsymbol{u} \tag{14}$$

Remarks

$$\lambda_{1,2} = 23 - \ln \left[\frac{Z_1 Z_2 (\mu_1 + \mu_2)}{\mu_1 T_2 + \mu_2 T_1} \left(\frac{n_1 Z_1^2}{T_1} + \frac{n_2 Z_2^2}{T_2} \right)^{1/2} \right] \quad (15)$$

Remarks

$$\lambda_{1,2} = 23 - \ln\left[\frac{Z_1 Z_2(\mu_1 + \mu_2)}{\mu_1 T_2 + \mu_2 T_1}\left(\frac{n_1 Z_1^2}{T_1} + \frac{n_2 Z_2^2}{T_2}\right)^{1/2}\right] \quad (15)$$

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (16)$$

$$x = \sigma\sqrt{-2\ln\gamma_1}\cos(2\pi\gamma_2) \quad (17)$$

Remarks

$$\lambda_{1,2} = 23 - \ln\left[\frac{Z_1 Z_2(\mu_1+\mu_2)}{\mu_1 T_2 + \mu_2 T_1}\left(\frac{n_1 Z_1^2}{T_1} + \frac{n_2 Z_2^2}{T_2}\right)^{1/2}\right] \tag{15}$$

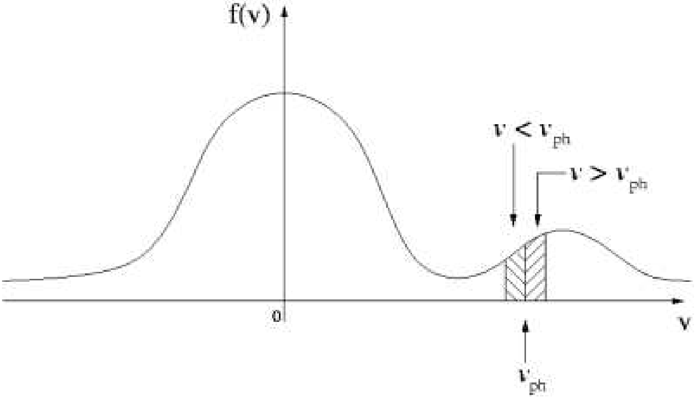
$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{16}$$

$$x = \sigma\sqrt{-2\ln\gamma_1}\cos(2\pi\gamma_2) \tag{17}$$

$$\sigma = \sqrt{\frac{k_\mathrm{B} T}{m}} \tag{18}$$

$$\langle v^2\rangle - \langle v\rangle^2 = \frac{k_\mathrm{B} T}{m}\left(3 - \frac{8}{\pi}\right) \tag{19}$$

Two Stream Instability



Simulation Set-up

Simulation Set-up

- Programming language for the 3D simulation - Compaq Visual Fortran Professional Edition 6.6.C

Simulation Set-up

- Programming language for the 3D simulation - Compaq Visual Fortran Professional Edition 6.6.C
- The spatial unlimited target plasma simulated through a cube with periodic boundary conditions

Simulation Set-up

- Programming language for the 3D simulation - Compaq Visual Fortran Professional Edition 6.6.C
- The spatial unlimited target plasma simulated through a cube with periodic boundary conditions
- The side of the cube 10^{-5}, 10^{5} nuclei of deuterium in target

Simulation Set-up

- Programming language for the 3D simulation - Compaq Visual Fortran Professional Edition 6.6.C
- The spatial unlimited target plasma simulated through a cube with periodic boundary conditions
- The side of the cube 10^{-5}, 10^{5} nuclei of deuterium in target
- The beam of 10^{4} deuterium particles with initial velocity 10^{3} kms^{-1}

Simulation Set-up

- Programming language for the 3D simulation - Compaq Visual Fortran Professional Edition 6.6.C
- The spatial unlimited target plasma simulated through a cube with periodic boundary conditions
- The side of the cube 10^{-5}, 10^{5} nuclei of deuterium in target
- The beam of 10^{4} deuterium particles with initial velocity 10^{3} kms^{-1}
- Maxwellian velocity distribution with temperature 1 keV

Simulation Set-up

- Programming language for the 3D simulation - Compaq Visual Fortran Professional Edition 6.6.C
- The spatial unlimited target plasma simulated through a cube with periodic boundary conditions
- The side of the cube 10^{-5}, 10^{5} nuclei of deuterium in target
- The beam of 10^{4} deuterium particles with initial velocity 10^{3} kms^{-1}
- Maxwellian velocity distribution with temperature 1 keV
- Time step 10^{-12} s

Simulation Set-up

- Programming language for the 3D simulation - Compaq Visual Fortran Professional Edition 6.6.C
- The spatial unlimited target plasma simulated through a cube with periodic boundary conditions
- The side of the cube 10^{-5}, 10^{5} nuclei of deuterium in target
- The beam of 10^{4} deuterium particles with initial velocity 10^{3} kms^{-1}
- Maxwellian velocity distribution with temperature 1 keV
- Time step 10^{-12} s
- Condition for the realization of a collision - empirically set the distance of $\lambda_{\mathrm{De}}/100$

Simulation Set-up

- Programming language for the 3D simulation - Compaq Visual Fortran Professional Edition 6.6.C
- The spatial unlimited target plasma simulated through a cube with periodic boundary conditions
- The side of the cube 10^{-5}, 10^{5} nuclei of deuterium in target
- The beam of 10^{4} deuterium particles with initial velocity 10^{3} kms^{-1}
- Maxwellian velocity distribution with temperature 1 keV
- Time step 10^{-12} s
- Condition for the realization of a collision - empirically set the distance of $\lambda_{\mathrm{De}}/100$
- Collisions in separated Maxwellians neglected, straightforward motion, temperature calculation

Simulation Results

Simulation Results

- After performance of 100 time steps - end of interactions

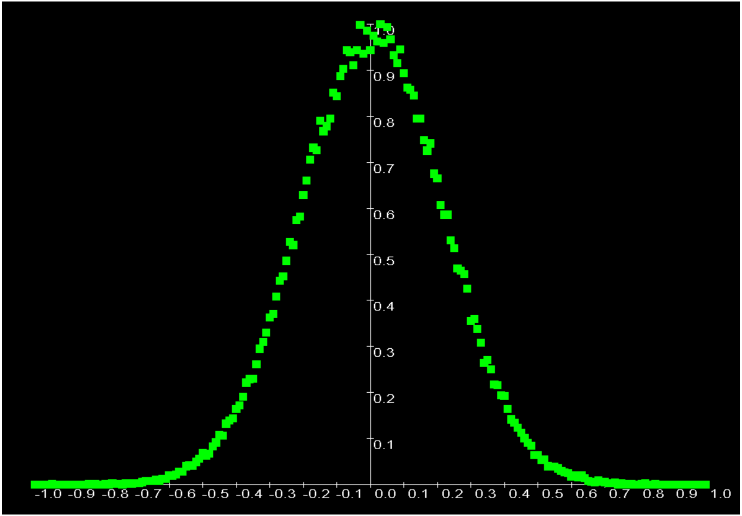
Simulation Results

- After performance of 100 time steps - end of interactions
- Agreement with the theory - kinetic energy of the beam transformed to the thermal energy both beam and target particles

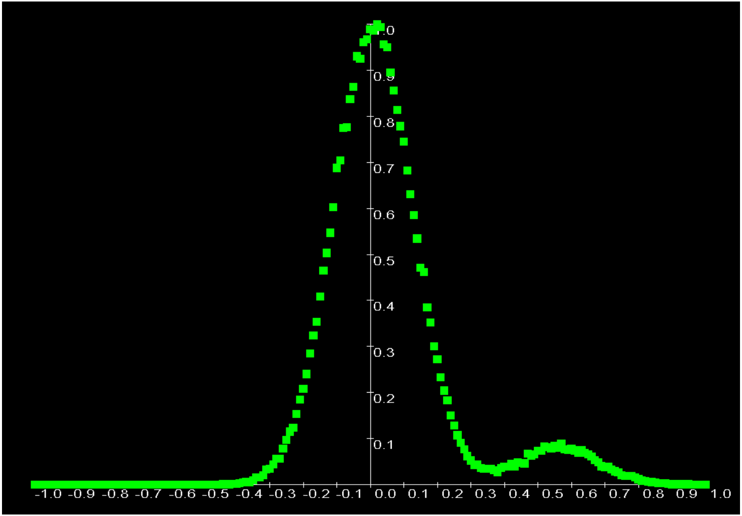
Simulation Results

- After performance of 100 time steps - end of interactions
- Agreement with the theory - kinetic energy of the beam transformed to the thermal energy both beam and target particles
- The beam heated up to 3 keV, the target to 2 keV

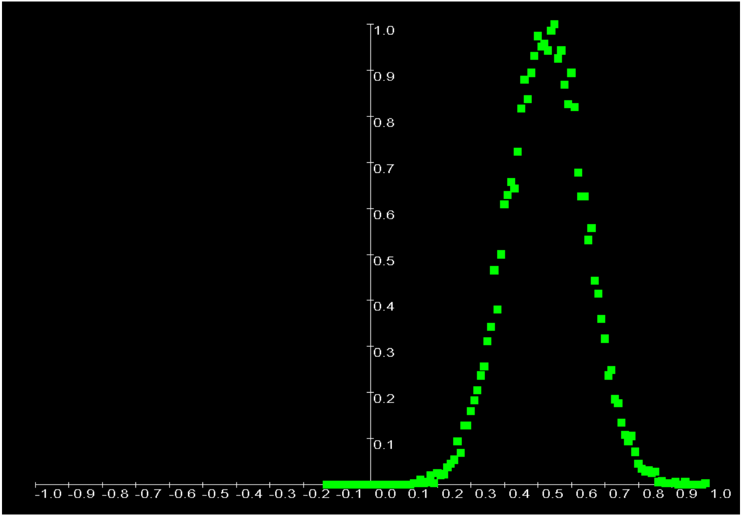
Initial target velocities distribution



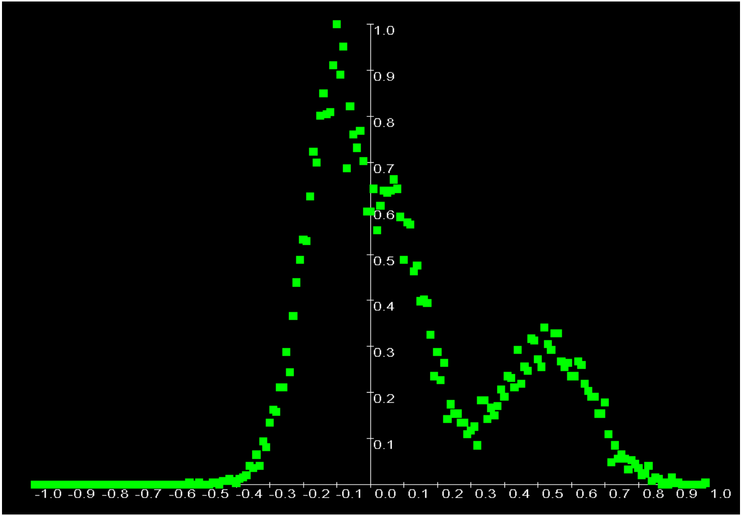
Final target velocities distribution



Initial beam velocities distribution



Final beam velocities distribution



Diploma Thesis

- Beam - target model

Diploma Thesis

- Beam - target model
- External magnetic field

Diploma Thesis

- Beam - target model
- External magnetic field
- A few experimental, theoretical, and computer results

Diploma Thesis

- Beam - target model
- External magnetic field
- A few experimental, theoretical, and computer results
- More interactions - elastic, ...

Diploma Thesis

- Beam - target model
- External magnetic field
- A few experimental, theoretical, and computer results
- More interactions - elastic, ...
- Better collision and boundary conditions