Biannual Progress Report - 2017.02.28

I-Lin Yeh 葉宜霖

Pulsed-power generator for space physics labratory, Institute of space and plasma physics, National Cheng Kung University, Taiwan

Adivisor: Dr. Po-Yu Chang 張博宇博士

Computer simulation is a good way to understand the detailed mechanisms of plasma behavior and the acquaintance with numerical methods is very important for doing this. Numerical hydrodynamics including solving ODE, parabolic PDE and linear advection equations were being practiced in last semeseter. Particle-in-cell (PIC) simulations will be studied in the next half year since I am going to develop a fully kinetic PIC code for studying the mechanisms of neutron production in Dense Plasma Focus device, which is a pulsed-power device used as an efficient neutron source. The results of this project will assist the design of the DPF device as the neutron source which will be built by the Institute of Nuclear Energy Research (INER) Atomic Energy Council through the expected cooperation between NCKU and INER.

Contents

1 Introduction **3**

1.1 Numerical hydrodynamics 3

1.2 Dense Plasma Focus 4

1.3 My research goal 6

2 List of works **7**

2.1 Practicing developing codes of numerical hydrodynamics 7

2.1.1 Numerical methods for solving simple ODE 7

2.1.2 Introduction to numerical methods for solving PDE 8

2.1.3 Numerical methods for solving parabolic PDE (heat equation) . . 10

2.1.4 Numerical methods for solving linear advection equation 12

2.2 Understanding the principles of particle-in-cell simulations 23

2.2.1 Particle-in-cell simulations 23

2.2.2 GPU accelerated particle-in-cell simulations 24

2.3 Paper reviews on simulations of pinch phase in DPF devices 25

3 Future works **28**

3.1 The 1^{st} and 2^{nd} steps 28

3.2 The 3^{rd} and 4^{th} steps 29

3.3 The 5^{th} step 29

4 Conclusion **30**

1 Introduction

Plasma is the fourth state of matter, comprised of ions, electrons and neutral atoms, usually at temperature above 10^4 degrees Kelvin. Plasma makes up most of the matter in the universe. It is also the state of the matter for the magnetic and inertial confinement fusion. Since plasma is a very complex system, the computer simulations of plasmas play a significant role in developing plasma theory. "Simulation is almost always the only direct experience we can have with detailed dynamics and behaviors of plasmas; almost all other experiences (in theory and lab) are indirect," quoted from the book "Plasma physics via computer simulations"[1]. Computer simulations can indeed give the clear view of every process in the complex system like plasma. There are two general areas in simulations of plasma based on kinetic and fluid descriptions, as shown in Figure 1[1]. Fluid simulation means numerically solving the magnetohydrodynamics (MHD) equations of plasma with the assumption of approximate transport coefficients. Kinetic simulations consider more detailed models of plasma. This can be attained either by numerically solving the plasma kinetic equations, e.g. Vlasov or Fokker-Plank equations or by particle-in-cell (PIC) simulations, which computes the motions of a collection of charged particles interacting with each other and with external applied fields.

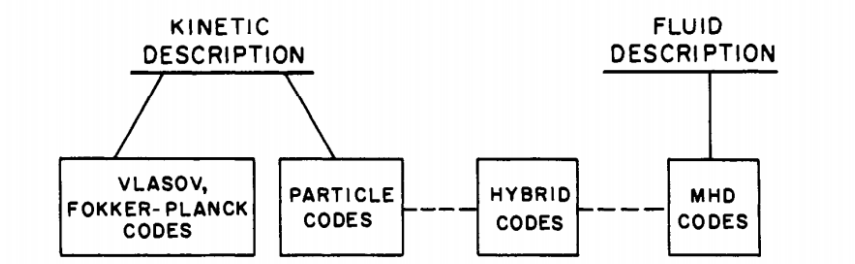

Figure 1: Classification of computer simulation models of plasmas.

1.1 Numerical hydrodynamics

My undergraduate research is most related to developing codes, so it is important for me to have a strong foundation of coding skills and plasma simulation. I learned numerical analysis through taking the class in the Aeronautic department this semester. After be-

ing familiar with numerical simulation more, applying the numerical methods to plasma physics is the next step. Numerical hydrodynamics is a good topic to understand the basic skills for solving partial differential equations; moreover, it is like a simple model of magnetohydrodnamics, which is one of the descriptions of plasma physics. Magnetohydrodynamics describes the dynamics of plasma in the fluid model. The only difference between magnetohydrodynamics and hydrodynamics is that magnetohydrodynamics considers the interaction of electromagnetic fields but hydrodynamics does not. I practiced numerical hydrodynamics following the book "Introduction to numerical hydrodynamics" [2]. The basic numerical skills for solving PDE were learned through practicing numerical hydrodynamics.

1.2 Dense Plasma Focus

Dense Plasma Focus is a kind of pinch discharge in which a pulsed voltage is applied to a low-pressure gas between coaxial cylindrical electrodes, generating a short duration, high density plasma region in the axis.[3] It was invented by Fillipov[4] and Mather[5] independently in 1960s and is still being studied intensely around the world as low-cost sources of neutrons, ions and x-ray as well as the potential fusion device.

Dense Plasma Focus device is a pulsed-power device comprised primarily of a coaxial configuration, where the inner electrode is anode and the outer is cathode. Figure 2[6] shows the 4 phases of DPF. (1) Plasma current sheaths are formed from the high voltage discharge making the flashover event occur along the surface of the insulator. (2) $\boldsymbol{j} \times \boldsymbol{B}$ forces provide lift-off of the current sheaths toward the end of the anode, which is called "lift-off" or "run-down". (3) As the current sheaths reach the anode tip, the plasma is accelerated radially inward or "run-in" and forms a dense z-pinch with particle densities of $\sim 10^{19}$ n/cm^3 and temperatures of ~ 100 eV, lasting tens of nanoseconds in the majority of devices. (4) In the pinch phase, the plasma implodes and creates a high-density region that typically emits ion beams, high-energy electrons, x-rays, and neutrons (in the presence of D or D-T gas). The whole process lasts a few microseconds.

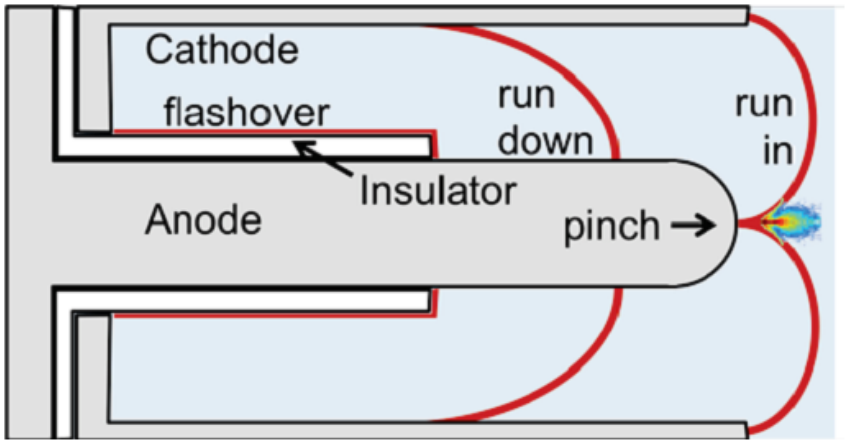

Figure 2: An illustration of the 4 phases of DPF: (1) initiation via flashover of the insulator; (2) plasma being accelerated to axial "run-down" phase; (3) radial implosion of "run-in" phase and (4) formation of a z-pinched plasma on axis.

There are two different kinds of DPF devices, differing in their electrode aspect ratio (electrode length over inner electrode diameter): the Filipov configuration[4], with an aspect ratio less than 1 (typical values are 0.2), and the Mather configuration[5], with an aspect ratio greater than 1 (typically 5-10). Dense Plasma Focus devices have been constructed in various sizes as neutron sources or other applications around the world in correlation with the energy stored in the pulsed-power generator, ranging from kilojoules to megajoules, producing average neutron pulses from 10^7 to 10^{12} neutrons per shot.

Though the mechanisms of plasma formation, neutron production and ion beam formation are still not completely understood, a variety of empirical, experimental and modeling information is available to optimize the design of DPF as neutron sources.[7, 3, 8] The engineering development of DPF has resulted in a mobile DPF neutron source with significantly higher peak and average neutron output compared with conventional neutron sources; for example, the work in Italy has already demonstrated a semi-mobile 6 kJ DPF filled with deuterium gas operating at average neutron output of 3×10^8 n/s with a repetition rate of 1 Hz, giving a peak neutron output $\sim 10^{15}$ n/s, which has the same or better performance compared to conventional accelerator-based neutron sources.[9] Dense Plasma Focus devices can be used not only as neutron sources but also particle beams and soft x-ray sources. The soft x-ray source can be applied to nanofabrication.[8]

Further, Lawrenceville Plasma Physics, Inc. (LPP fusion), a company using DPF as the fusion device to produce energy, announced that they achieved two of the three conditions required to fuse hydrogen-boron and produce net energy in 2012.[1] Plasma temperature of 150 keV and sufficient confinement time were achieved.[10] If the results claimed by Lerner, the president of LPP fusion, are right, it means that DPF can be a feasible fusion device. Since DPF has so many applications, it is essential to understand the detailed mechanisms of it.

1.3 My research goal

My research topic is to use particle-in-cell (PIC) model via computing parallelly using GPU to simulate the process of neutron production through the final z–pinch phase and finally get the neutron yield. Since Institute of Nuclear Energy Research (INER) Atomic Energy Council (行政院原子能委員會核能研究所) is building DPF device as a neutron source, the results of this project can be used as a potential cooperation with INER. Furthermore, the optimization of the neutron production of the DPF device will no longer be done by trial-and-error but by the fully understood mechanisms in future.

This report shows the work on developing codes of numerical hydrodynamics as well as understanding the principle of PIC simulations and then the studies of paper reviews on simulation of pinch phase in DPF device. The procedures and progress are shown in the final part. The ultimate goal for me is to finish my research topic as described above and understand plasma physics and programming thoroughly.

[1] http://lppfusion.com/executive-summary/

2 List of works

This part shows the work I did in last half year, including (1) developing codes of numerical hydrodynamics; (2) understanding the principle of PIC simulations; (3) paper reviews on simulaiton of pinch phase in DPF device.

2.1 Practicing developing codes of numerical hydrodynamics

2.1.1 Numerical methods for solving simple ODE

Discretization The simple ordinary differential equation (ODE)

$$\frac{dy}{dx} = -a\,y \tag{1}$$

with initial value $y(t_0) = y_0$ is of first order and linear with constant coefficient a. It has the obvious analytical solution

$$y(t) = y_0\, e^{-a(t-t_0)}. \tag{2}$$

For discrete time steps

$$t^n = \Delta t\, n + t_0, \tag{3}$$

using explicit Euler scheme (approximation of y by a piecewise linear curve) can give Eq. (1) the most simple discretization

$$y^{n+1} = y^n - a\, y^n\, \Delta t. \tag{4}$$

Examples Figure 3 shows examples of Eq. (4) with $a = 1$ for Δt = 2.20, 1.70, 0.90, 0.45, 0.01. Too large time steps lead to conflicts with the analytical solutions. The important properties for a good numerical scheme is as follows:

Criterion for stability (discrete operator does not amplify noise): $\Delta t < 2/a$,

Criterion for positivity (dependent variable is always positive): $\Delta t < 1/a$.

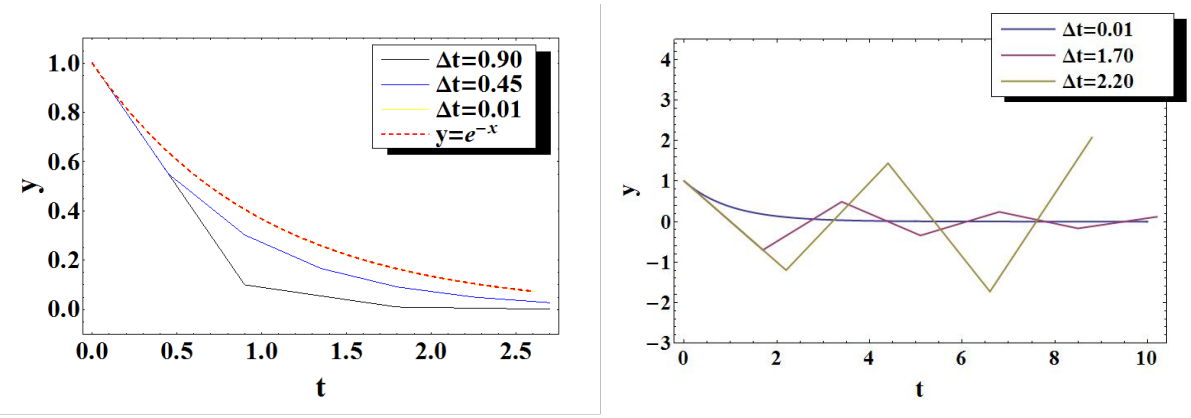

Figure 3: Examples of using explicit Euler scheme to solve ODE with different time steps.

2.1.2 Introduction to numerical methods for solving PDE

Stencil diagram To solve PDE numerically, space and time of dependent variables will be discretized, where the subscript stands for temporal grid and the superscript stands for spatial grid. Stencil diagram tells that the density ρ_i^{n+1}at spatial grid i and time step $n+1$ depends on which values at spatial grid and the old time-step n, e.g. ρ_{i-k}^n, $\rho_{i-k+1}^n, \ldots, \rho_{i-1}^n$. For example, the stencil diagram in Figure 4 shows that ρ_i^{n+1} depends on ρ_{i-2}^n, ρ_{i-1}^n, ρ_i^n, ρ_{i+1}^n, ρ_{i+2}^n.

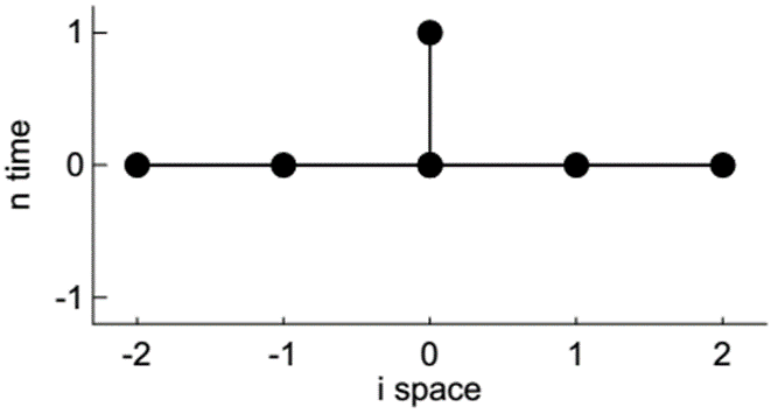

Figure 4: Stencil diagram. (Horizontal axis is temporal grid; vertical axis is spatial grid)

Boundary conditions Boundary conditions influence the real-world situations of hydrodynamic flows. For instance, the dependent variables like the density at time step

n physically discussed are ρ_1^n, ρ_2^n,. . . , ρ_{10}^n. According to the stencil diagrams for different numerical schemes, the density ρ_1^n and ρ_{10}^n at boundary spatial grids and time step n may depend on the density at the spatial grids beyond physically discussed grids and time step $n-1$ (e.g. $\rho_0^{n-1}$and ρ_{11}^{n-1}), so the value of density at the spatial grids beyond physically discussed grids called "Ghost cell" will be needed, as shown in Figure 5.

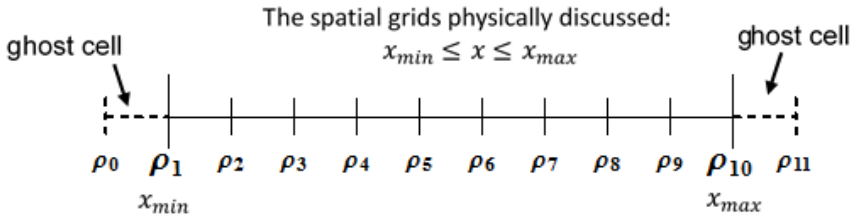

Figure 5: A schematic for ghost cell and boundary grids. (Superscripts of the densities ρ for temporal grids are ignored for convenience.)

There are three common boundary conditions. Using stencil diagram in Figure 4 for example, so ghost cells ρ_{-1}, ρ_0, ρ_{11}, ρ_{12} are needed.

1. Neumann boundary condition ($\frac{\partial \rho}{\partial x}|_{boundary} = 0$):

$$\rho_{-1} = \rho_1,\ \rho_0 = \rho_1;\ \rho_{11} = \rho_{10},\ \rho_{12} = \rho_{10}.$$

2. Periodic boundary condition (the values of ρ at the two sides of spatial boundary grids are the same):

$$\rho_{-1} = \rho_8,\ \rho_0 = \rho_9;\ \rho_{11} = \rho_2,\ \rho_{12} = \rho_3.$$

3. Reflective boundary condition (the values of ρ are symmetric about the spatial boundary grids):

$$\rho_{-1} = \rho_3,\ \rho_0 = \rho_2;\ \rho_{11} = \rho_9,\ \rho_{12} = \rho_8.$$

2.1.3 Numerical methods for solving parabolic PDE (heat equation)

Discretization The heat equation

$$\frac{\partial y}{\partial t} = K \frac{\partial^2 y}{\partial x^2} \tag{5}$$

is a parabolic PDE with constant coefficient K and y is a function of x and t. As time goes by, the tendency for the peaks to be eroded and the valley to be filled will be observed as expected. For discrete time steps and spatial grids

$$t^n = \Delta t\, n + t_0;\ x = \Delta x\, i + x_0, \tag{6}$$

the first-order time derivative of y and second-order space derivative of y can be expressed discretely as

$$\frac{\partial y}{\partial t} \rightarrow \frac{y_i^{n+1} - y_i^n}{\Delta t}, \tag{7}$$

$$\frac{\partial^2 y}{\partial x^2} \rightarrow \frac{y_{i+1}^n - 2\, y_i^n + y_{i-1}^n}{\Delta x^2}. \tag{8}$$

Thus, it gives the explicit Euler scheme

$$y_i^{n+1} = y_i^n + \frac{\Delta t}{\Delta x^2} K\, (y_{i+1}^n - 2\, y_i^n + y_{i-1}^n). \tag{9}$$

Examples for jagged initial condition Examples for jagged (1, 0, 1, 0, 1, ...) initial condition (red dashed line) with periodic boundary condition are displayed in Figure 6.When the grid settings satisfy the criterion for stability $\frac{\Delta t\, K}{\Delta x^2} < 0.5$, the numerical results (blue line) for heat equation will be stable.

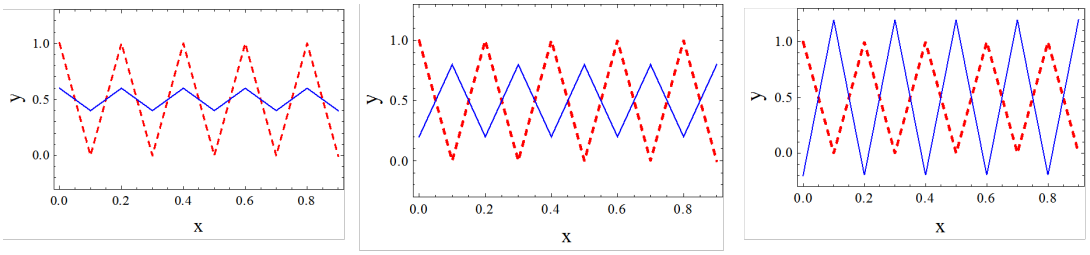

Figure 6: Numerical results for heat equation for jagged initial condition with grid setting: $\frac{\Delta t\, K}{\Delta x^2} = 0.2,\ 0.4,\ 0.6$ are shown respectively from left to right.

Examples for spike initial condition Examples for spike initial condition (red dashed line) with periodic boundary condition are displayed in Figure 7. The spike initial condition is defined as Gaussian, rectangle, triangle, super Gaussian(exp(-x^4)) respectively from left to right. The criterion for stability is $\frac{\Delta t\, K}{\Delta x^2} < 0.5$. The grid setting $\frac{\Delta t\, K}{\Delta x^2} > 0.5$ will cause the spurious oscillation at the extremely big-gradient grids, e.g. rectangle, in the right panel.

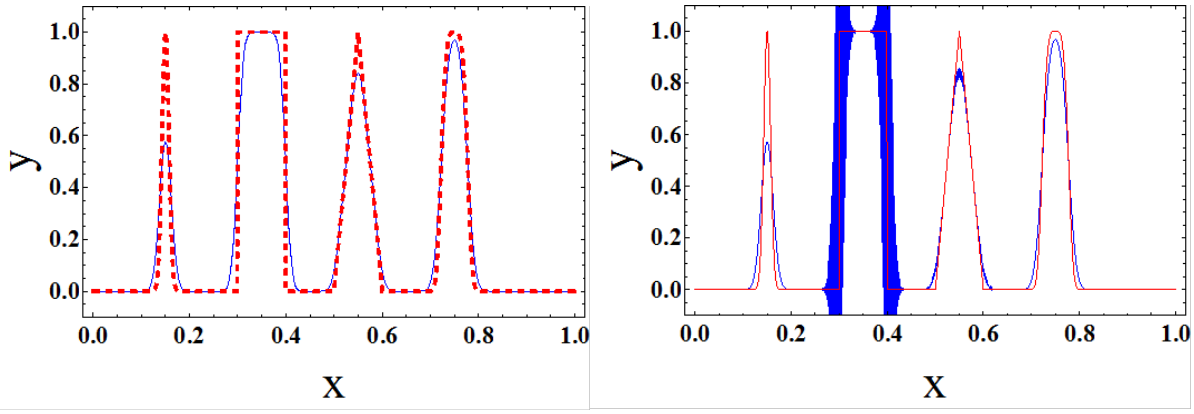

Figure 7: Numerical results for heat equation for spike initial conditions with grid setting: $\frac{\Delta t\, K}{\Delta x^2} = 0.5,\ 0.51,$ are shown respectively from left to right.

Examples for Gaussian initial condition Heat equation with Gaussian initial condition

$$y(x,\, 0) = exp(-(\frac{x - 0.5}{0.1})^2) \tag{10}$$

can be solved analytically by Fourier transform. The analytical solution of heat equation with Gaussian initial condition at time t is

$$y(x,\,t) = \sqrt{\frac{1}{1+400\,t}}exp(-(\frac{x-0.5}{\sqrt{0.01+4\,t}})^2). \tag{11}$$

Take $t = 0.01$, we can get

$$y(x,\,0.01) = \sqrt{\frac{1}{5}}exp(-(\frac{x-0.5}{\sqrt{0.05}})^2). \tag{12}$$

The comparison of analytical results and numerical results, which is used the same initial condition as analytical results, is displayed in Figure 8. Since there exists analytical solution for heat equation with Gaussian initial condition, heat equation with Gaussian initial condition can be a benchmark for checking whether the numerical methods is viable.

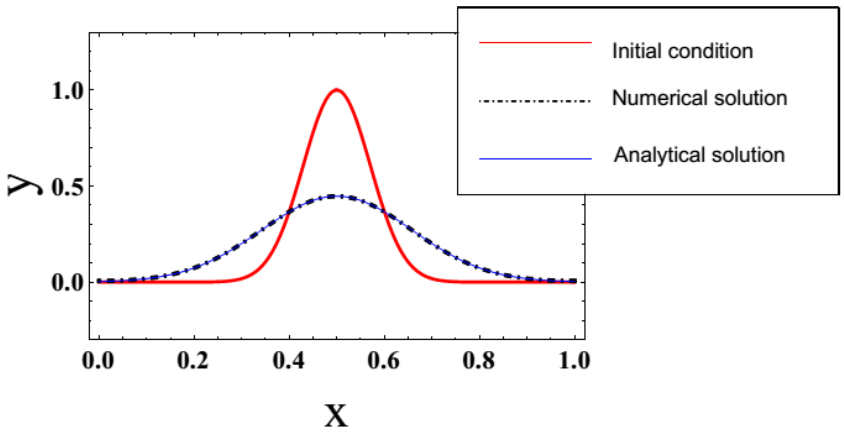

Figure 8: Numerical results and analytical results for heat equation with Gaussian initial condition are shown.

2.1.4 Numerical methods for solving linear advection equation

Discretization Linear advection equation (conservation of mass)

$$\frac{\partial\,\rho}{\partial\,t} + v\,\frac{\partial\,\rho}{\partial\,x} = 0, \tag{13}$$

where $\rho = \rho(x,\,t)$ and v is velocity, has analytical solution $\rho(x,\,t) = \rho(x - v\,t)$. Using periodic boundary condition, function ρ will keep propagating forwardly (when an object passes through one side of the figure, it re-appears on the opposite side with the same

velocity). We can take t =one period ($T = \frac{x_{max}-x_{min}}{v}$) and the solution will be same as the initial condition, which means $\rho(x,\, T) = \rho(x,\, 0)$. Thus ,the resemblance between initial condition and numerical result means this numerical scheme is suitable for this condition.

To solve linear advection equation numerically, we discretize the equation using the grid setting as below:

$$\Delta x = \frac{1}{i_{total}} = \frac{1}{200},\ v = 1,\ \Delta t = 0.002,\ \frac{\Delta t\, v}{\Delta x} = 0.4,\ 500\, time\ steps\, (1T).$$

Spike initial condition and periodic boundary condition will be used. We define the fluxes in the cells

$$f(\rho_i^n) \equiv v\, \rho_i^n$$

and the fluxes at cell boundaries

$$f_{i+\frac{1}{2}}^n \equiv v\, \rho_{i+\frac{1}{2}}^n$$

for simplify the linear advection equation, the equation can be rewritten in the form as

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x}\,(f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n). \tag{14}$$

Linear advection equation as the function of fluxes characterizes the physical meaning that the densities in the cell will be influenced by the fluxes at the boundaries. Accordingly, the following will show the numerical results of different schemes with different definitions of fluxes for solving linear advection equation.

Forward time central space (FTCS) scheme Flux for FTCS scheme is defined as

$$f_{i+\frac{1}{2}}^n = \frac{1}{2}\,[f(\rho_{i+1}^n) + f(\rho_{i+1}^n)], \tag{15}$$

so Eq. 14 can be updated as below

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{2\,\Delta x}\, v\,(\rho_{i+1}^n - \rho_{i-1}^n). \tag{16}$$

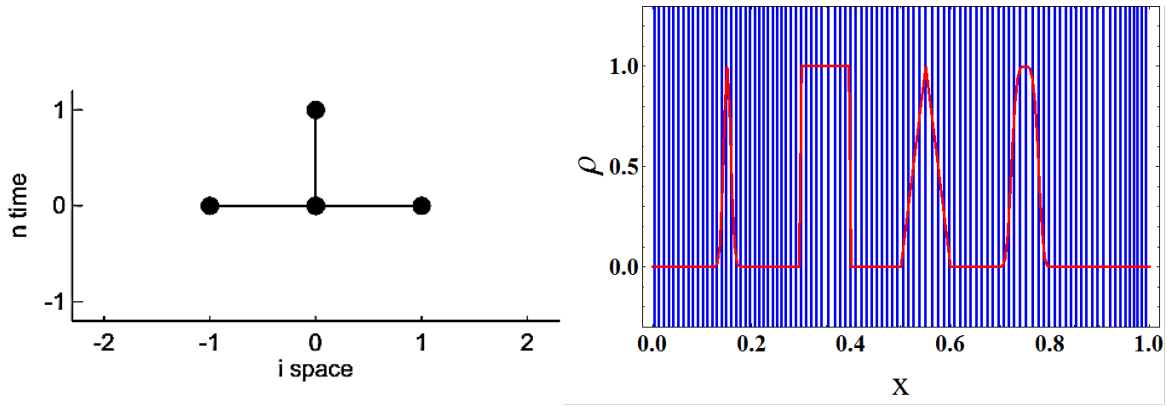

Figure 9: FTCS scheme: stencil diagram and test result (red dashed line as true solution and blue line as numerical result).

Numerical result is shown in Figure 9. We can observe that it diverges seriously and doesn't have any resemblance to true solution.

Forward time backward space (FTBS) scheme Flux for FTBS scheme is defined as

$$f^n_{i+\frac{1}{2}} = f(\rho^n_i), \tag{17}$$

so Eq. 14 can be updated as below

$$\rho^{n+1}_i = \rho^n_i - \frac{\Delta t}{\Delta x} v\,(\rho^n_i - \rho^n_{i-1}). \tag{18}$$

Numerical result is shown in Figure 10. We can observe that it is a stable solution but diffuses seriously.

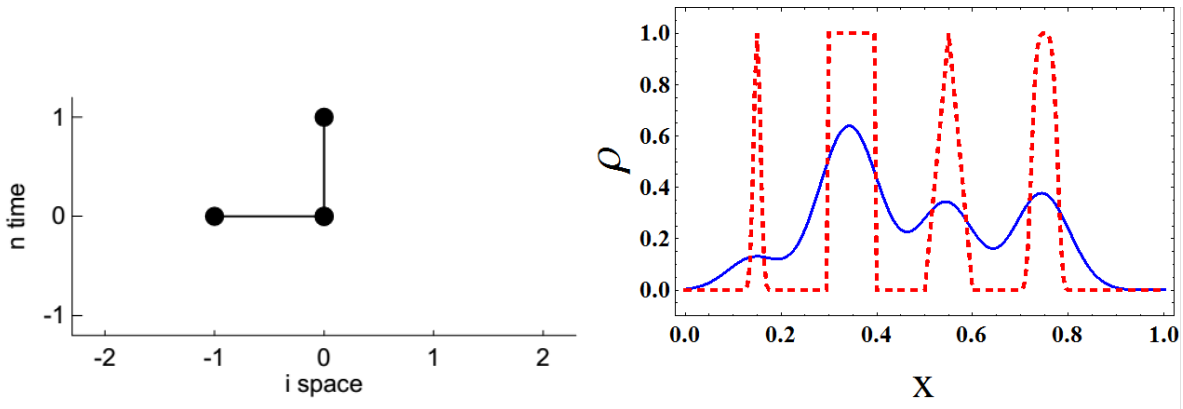

Figure 10: FTBS scheme: stencil diagram and test result (red dashed as true solution and blue line as numerical result).

Forward time forward space (FTFS) scheme Flux for FTBS scheme is defined as

$$f^n_{i+\frac{1}{2}} = f(\rho^n_{i+1}), \tag{19}$$

so Eq. 14 can be updated as below

$$\rho^{n+1}_i = \rho^n_i - \frac{\Delta t}{\Delta x} v\,(\rho^n_{i+1} - \rho^n_i). \tag{20}$$

Numerical result is shown in Figure 11. We can observe that it diverges seriously and doesn't have any resemblance to true solution.

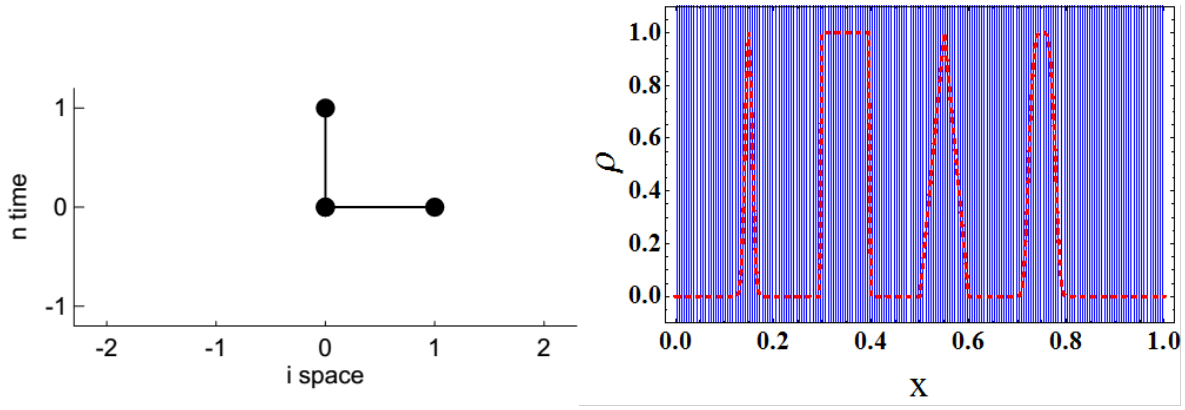

Figure 11: FTFS scheme: stencil diagram and test result (red dashed line as true solution and blue line as numerical result).

Lax-Friedrichs scheme Flux for Lax-Friedrichs scheme is defined as

$$f^n_{i+\frac{1}{2}} = \frac{1}{2}\left[f(\rho^n_{i+1}) + f(\rho^n_i)\right] - \frac{1}{2}\frac{\Delta x}{\Delta t}\,(\rho^n_{i+1} - \rho^n_i), \tag{21}$$

so Eq. 14 can be updated as below

$$\rho^{n+1}_i = \rho^n_i - \frac{\Delta t}{2\Delta x}\left[v\,(\rho^n_{i+1} - \rho^n_{i-1}) - \frac{\Delta x}{\Delta t}(\rho^n_{i+1} - 2\rho^n_i + \rho^n_{i-1})\right]. \tag{22}$$

Numerical result is shown in Figure 12. We can observe that the diffusive effect is so strong and there are some small-scale wiggles left.

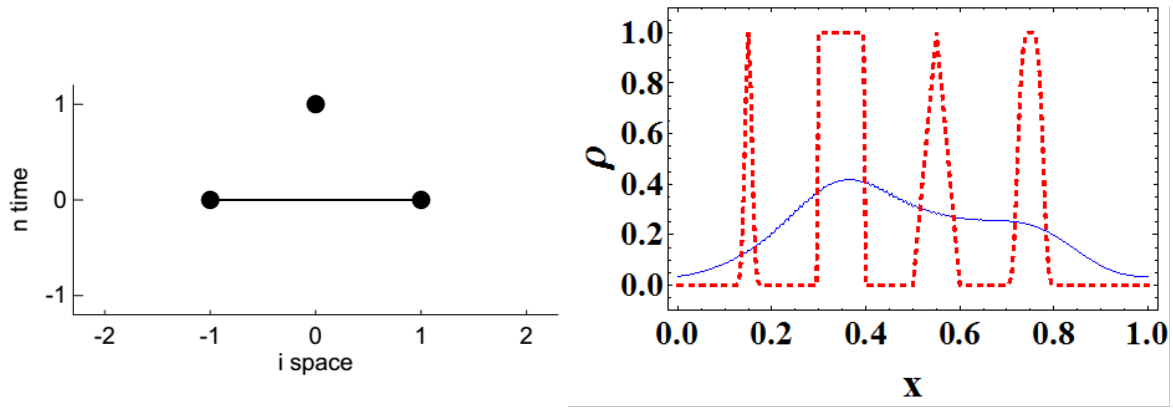

Figure 12: Lax-Friedrichs scheme: stencil diagram and test result (red dashed line as true solution and blue line as numerical result).

Gudunov-type finite-volume scheme Gudunov's idea first considers the evolution equation without any approximation by integrate Eq. 14 over one time step:

$$\rho_i^{n+1} = \rho_i^n - \frac{1}{\Delta x}\left\{\int_{t^n}^{t^{n+1}} f(\rho(x_{i+\frac{1}{2}},\, t))\, dt - \int_{t^n}^{t^{n+1}} f(\rho(x_{i-\frac{1}{2}},\, t))\, dt\right\}. \tag{23}$$

There are three procedures in Gudunov's idea: Reconstruct-Solve-Average (RSA): [11, 12]

1. Reconstruct: reconstruct a continuous function $\rho(x)$ from the discrete ρ_i.
2. Solve: form the exact right hand side time integrals and solve the exact problem for Δt.
3. Average: compute the cell averages for the new time level.

In the "Reconstruct" procedure, use piecewise linear method (PLM) to reconstruct a linear continuous function :

$$\rho(x,\, t) = \rho_i^n + (x - x_i)\, \sigma_i^n, \tag{24}$$

where σ_i^n is the slope at ith spatial grid and nth temporal grid.

In the "Solve" procedure, use the reconstructed function to compute the right-hand side integral:

$$
\begin{aligned}
&\int_{t^n}^{t^{n+1}} f(\rho(x_{i+\frac{1}{2}},\, t))\, dt = \Delta t\, f^n_{i+\frac{1}{2}} \\
=&\int_{t^n}^{t^{n+1}} f(\rho(x_{i+\frac{1}{2}} - v\,(t-t^n)))\, dt \\
=&\int_{t^n}^{t^{n+1}} v\, \left\{\rho^n_i + \left[x_{i+\frac{1}{2}} - v\,(t-t^n) - x_i\right] \sigma^n_i\right\} dt \\
=&\Delta t\, \left[v\,(\rho^n_i + \frac{\Delta x}{2}\,\sigma^n_i - \frac{v\,\Delta t}{2}\,\sigma^n_i)\right].
\end{aligned}
\tag{25}
$$

Hence, we can get the form of $f^n_{i+\frac{1}{2}}$ and $f^n_{i+\frac{1}{2}}$:

$$
f^n_{i+\frac{1}{2}} \equiv v\,(\rho^n_i + \frac{\Delta x}{2}\,\sigma^n_i - \frac{v\,\Delta t}{2}\,\sigma^n_i); \tag{26}
$$

$$
f^n_{i-\frac{1}{2}} \equiv v\,(\rho^n_{i-1} + \frac{\Delta x}{2}\,\sigma^n_{i-1} - \frac{v\,\Delta t}{2}\,\sigma^n_{i-1}). \tag{27}
$$

Plug fluxes at the cell boundaries into Eq. 14 and we finish the "Average" procedure:

$$
\begin{aligned}
\rho^{n+1}_i &= \rho^n_i - \frac{\Delta t}{\Delta x}\,(f^n_{i+\frac{1}{2}} - f^n_{i-\frac{1}{2}}) \\
&= \rho^n_i - \frac{\Delta t}{\Delta x}\,\left[v\,(\rho^n_i - \rho^n_{i-1}) + (\frac{v\,\Delta x}{2} - \frac{v^2\,\Delta t}{2})(\sigma^n_i - \sigma^n_{i-1})\right].
\end{aligned}
\tag{28}
$$

The followings are the examples of the different definition of slope σ^n_i((1) $\sim$ (4) are linear slope limiters and (5) $\sim$ (7) are high-resolution slope limiters):

(1) Forward time backward space (upwind)

The definition of slope is

$$
\sigma^n_i = 0. \tag{29}
$$

The result is shown in Figure 10.

(2) Downwind slope limiter (Lax-Wendroff scheme)

The definition of slope is

$$\sigma_i^n = \frac{\rho_{i+1}^n - \rho_i^n}{\Delta x}. \tag{30}$$

The result is shown in Figure 13. We can observe that it is smooth with strong overshoot. This numerical scheme is more suitable for smooth initial condition.

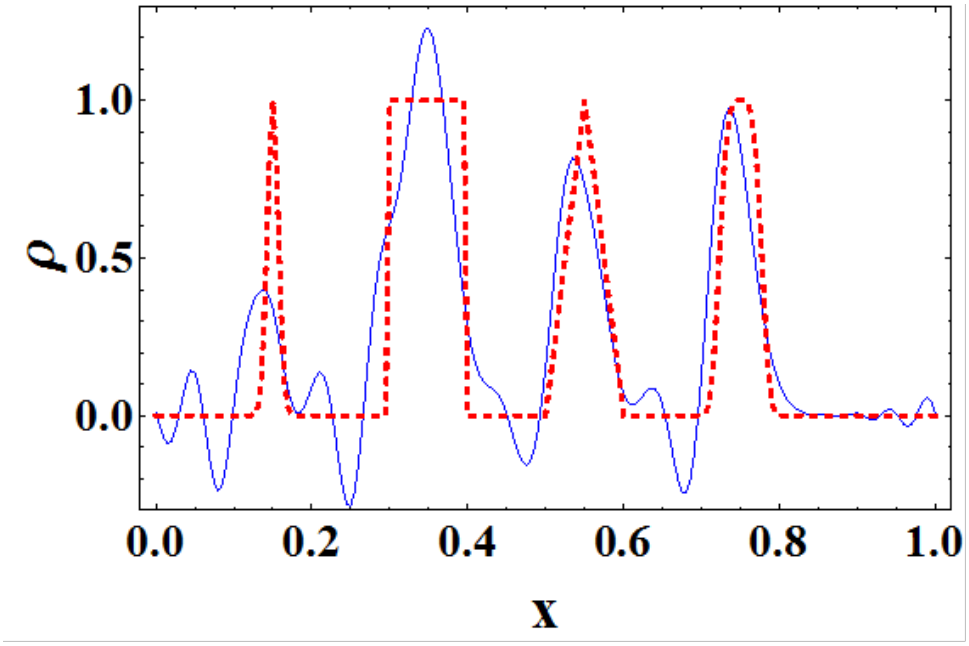

Figure 13: Numerical result for Lax-Wendroff scheme (red dashed line as true solution and blue line as numerical result).

(3) Upwind slope limiter (Lax-Warming scheme)

The definition of slope is

$$\sigma_i^n = \frac{\rho_i^n - \rho_{i-1}^n}{\Delta x}. \tag{31}$$

The result is shown in Figure 14. We can observe that it is smooth with strong overshoot. This numerical scheme is more suitable for smooth initial condition.

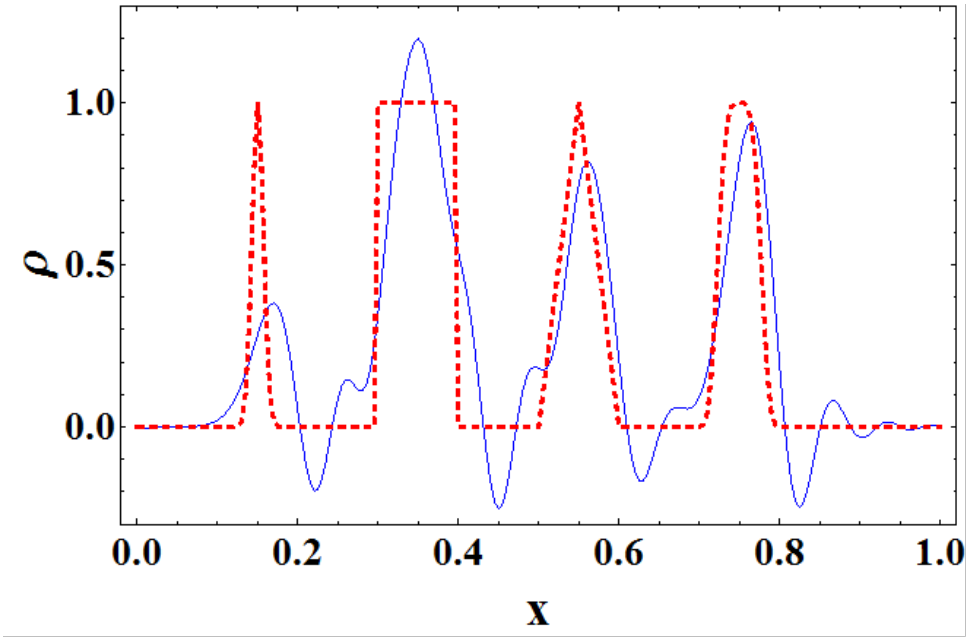

Figure 14: Numerical result for Lax-Warming scheme (red dashed line as true solution and blue line as numerical result).

(4) Centered slope limiter (Fromm scheme)

The definition of slope is

$$\sigma_i^n = \frac{\rho_{i+1}^n - \rho_{i-1}^n}{2\Delta x}. \tag{32}$$

The result is shown in Figure 15. We can observe that it is smooth with little overshoot but the initial shape of spike is still recognizable.

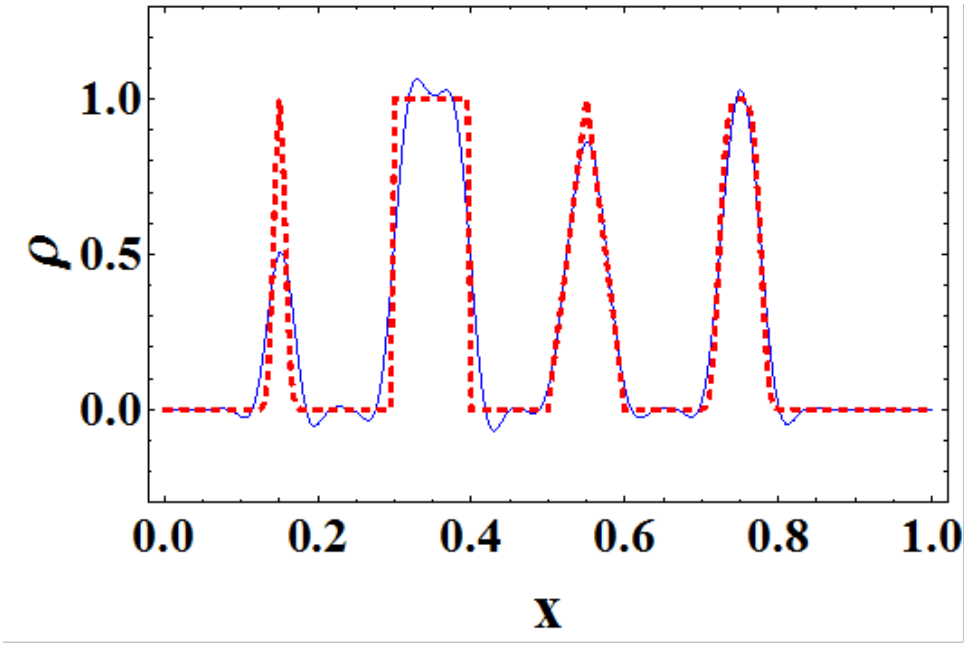

Figure 15: Numerical result for Lax-Warming scheme (red dashed line as true solution and blue line as numerical result).

(5) Mimnod slope limiter

The definition of slope (minimum allowed 2nd order slope) is

$$\begin{aligned}\sigma_i^n =& \frac{1}{\Delta x}\left[min(max(\rho_i^n - \rho_{i-1}^n, 0), max(\rho_{i+1}^n - \rho_i^n, 0))\right] \\ &+ \frac{1}{\Delta x}\left[max(min(\rho_i^n - \rho_{i-1}^n, 0), min(\rho_{i+1}^n - \rho_i^n, 0))\right]\end{aligned}. \tag{33}$$

The result is shown in Figure 16. We can observe that it is smooth with no overshoot and the initial shape of spike is still recognizable.

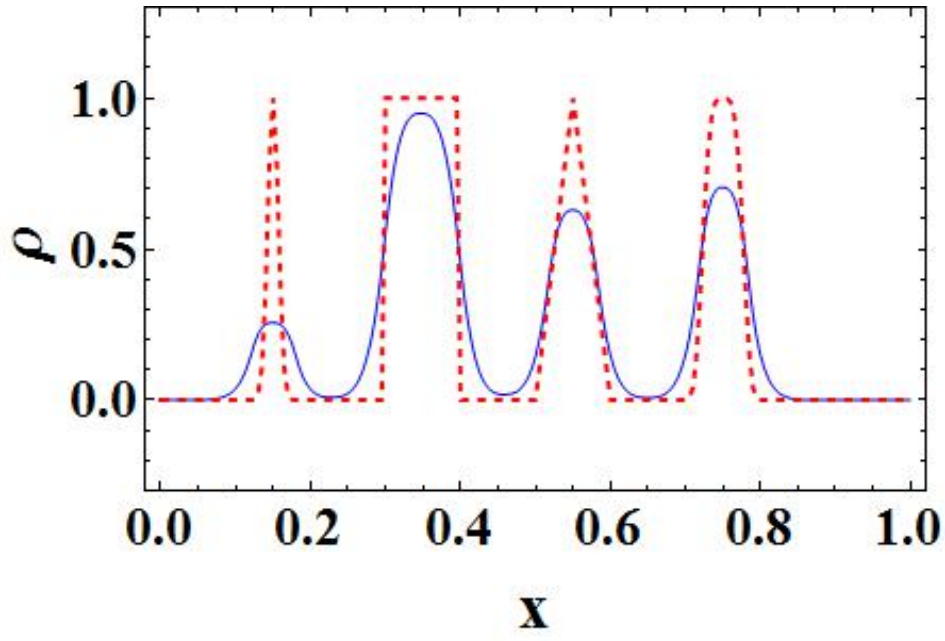

Figure 16: Numerical result for PLM scheme with Minmod slope limiter (red dashed line as true solution and blue line as numerical result).

(6) VanLeer slope limiter

The definition of slope (harmonic mean of slope) is

$$\sigma_i^n = \begin{cases} \frac{1}{\frac{\Delta x}{2}\left(\frac{1}{\rho_i^n - \rho_{i-1}^n} + \frac{1}{\rho_{i+1}^n - \rho_i^n}\right)} & if\ (\rho_i^n - \rho_{i-1}^n)\,(\rho_{i+1}^n - \rho_i^n) > 0 \\ 0 & elsewhere \end{cases}. \tag{34}$$

The result is shown in Figure 17. We can observe that it is smooth with no overshoot and the initial shape of spike is still recognizable.

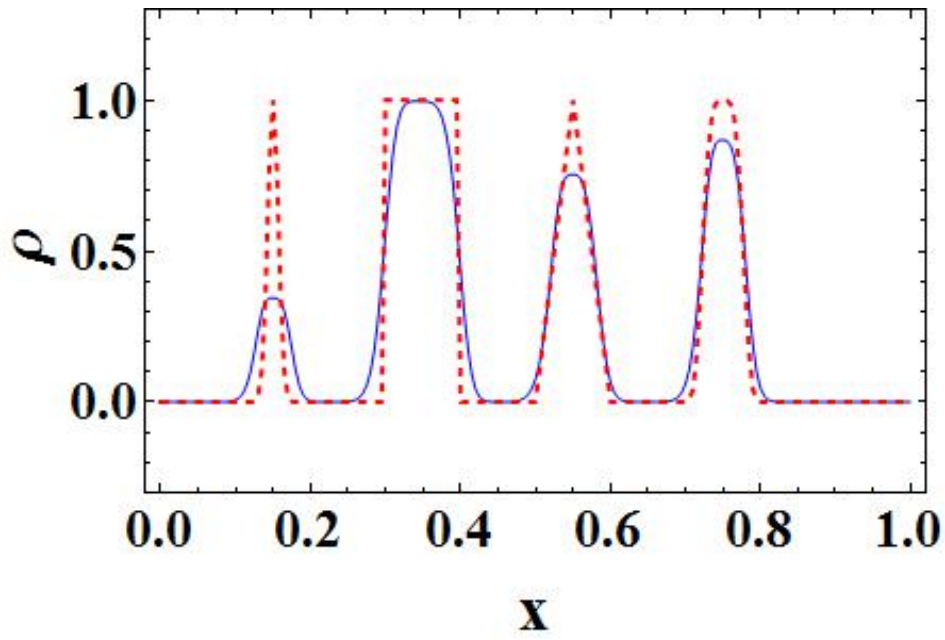

Figure 17: Numerical result for PLM scheme with VanLeer slope limiter (red dashed line as true solution and blue line as numerical result).

(7) Superbee slope limiter

The definition of slope (maximum allowed 2nd order slope) is

$$\sigma_i^n = \frac{1}{\Delta x}\left[sign(\rho_i^n - \rho_{i-1}^n) + sign(\rho_{i+1}^n - \rho_i^n)\right] \\ min\left(abs(\rho_i^n - \rho_{i-1}^n),\, abs(\rho_{i+1}^n - \rho_i^n),\, \frac{1}{2}\, max(abs(\rho_i^n - \rho_{i-1}^n), abs(\rho_{i+1}^n - \rho_i^n))\right). \tag{35}$$

The result is shown in Figure 18. We can observe that it is smooth with no overshoot and the initial shape of spike is still recognizable.

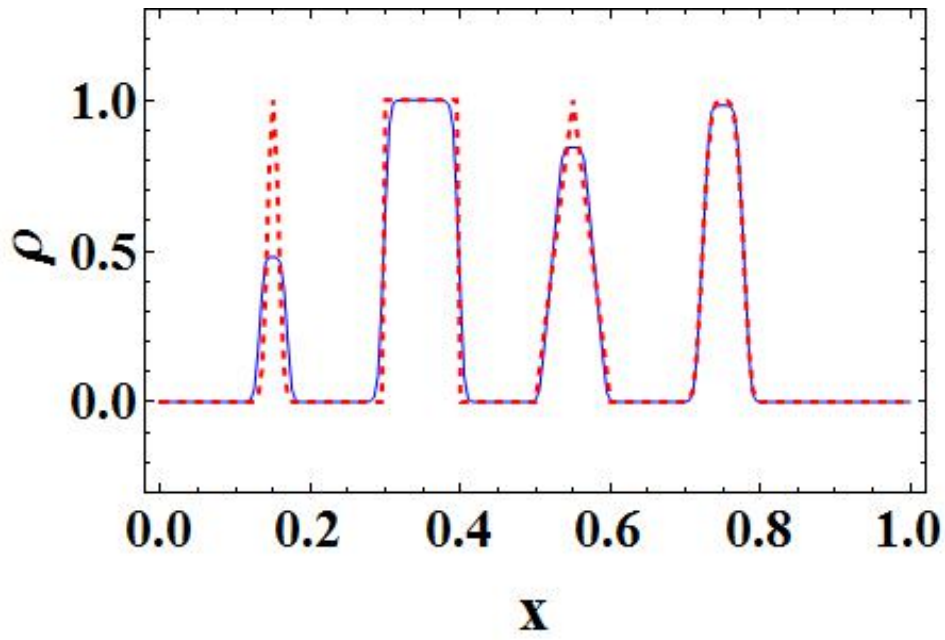

Figure 18: Numerical result for PLM scheme with Superbee slope limiter (red dashed line as true solution and blue line as numerical result).

2.2 Understanding the principles of particle-in-cell simulations

The pinch phase in DPF will be modeled by particle-in-cell (PIC) simulations, which is the first principle model without any physics approximations. Since PIC simulations would need intense computations, the parallel computing via Graphics Processing Unit (GPU) will be needed.

2.2.1 Particle-in-cell simulations

Particle-in-cell model is a first-principle model providing a kinetic description of a plasma without any physics approximations by following the trajectories of a collection of particles interacting with self-consistent electromagnetic fields. Each computational particle represents a certain number of real particles. The computational cycle of PIC method is shown in figure 19[13]. The basic description of PIC simulations is as following: (1) Maxwell equations are numerically solved on grids using the particle sources accumulated from the continuous particle locations to the discrete grids; (2) the particles are advanced one time step to new momentum and positions by numerically solving the equations of motion, using electromagnetic field interpolated from the discrete grid to

continuous particle locations; (3) loss or gain of particles at the boundary are considered; (4) if the model is collisional, Monte Carlo collisions of motion must also be considered; (5) the computational loop repeats and repeats.

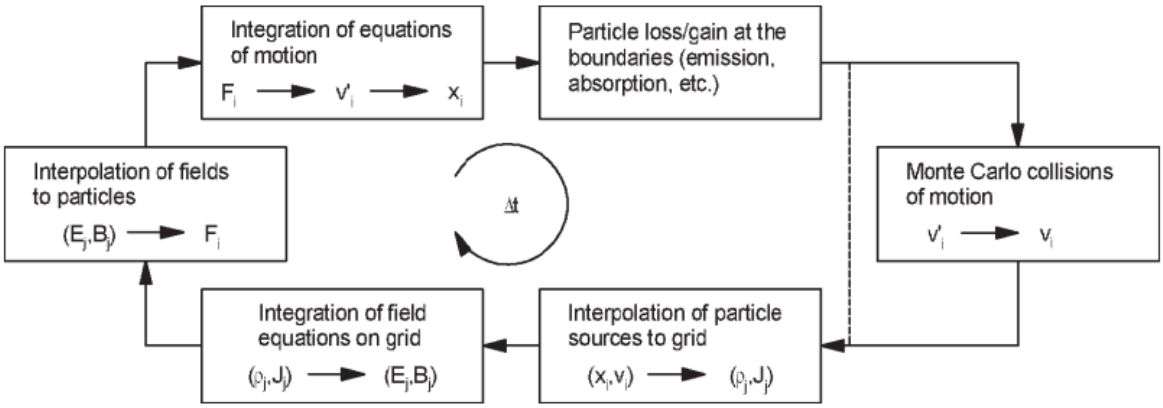

Figure 19: Basic flow loop of PIC simulations.

The program will simulate some test problems for benchmark. For example[1]: (1) checking the frequency of a simple harmonic motion of a pair of test electrons in a uniform background; (2) checking the growth rate as a function of wave vector in the instability of two opposing electron streams in a uniform background.

2.2.2 GPU accelerated particle-in-cell simulations

Particle-in-cell simulations are an important first-principle method to study plasma physics, but they require intense computations. Paralleling the PIC code can accelerate the computational process. Computing via GPU is an adequate way to do massive parallel data processing with arithmetic intensity. Considering a significant amount of particles, $\sim 10^8$ used in PIC simulations, the intense computation is appropriate to be executed on GPU. NVIDIA CUDA[14] has been a widely used GPU platform since its features such as easy programing and various applications. Thus, writing PIC code accelerated by GPU on CUDA has gradually became a new way to study the detailed kinetic behavior of plasma physics.[15]

2.3 Paper reviews on simulations of pinch phase in DPF devices

Significant amount of empirical results and scaling laws for the processes of DPF have been collected over 40 years, but the detailed mechanisms remain unclear nowadays especially in the final pinch phase. The mechanisms of the neutron production and the formation of high-energy ion beam are still an open and controversial question. In general, there are two ways of neutron production that are accepted widely: thermonuclear fusion and beam-target fusion. Thus, the neutron output can be divided into isotropic and anisotropic neutron emissions corresponding to thermonuclear component and beam-target component, respectively. Though the scaling laws from the experimental results for the neutron output in relation to peak current have been proposed by various groups[3], the detailed mechanisms are still unknown.

Magnetohydrodynamics (MHD) simulations can predict the process of coaxial run-down phase and the early part of the radial run-in phase nicely. However, current sheath formation and pinch phase are necessary to be simulated using kinetic model due to the occurrence of the instabilities and non-Maxwellian distribution function of particles. Collisions between neutral particles and electrons in sheath formation phase must be included in simulations so that kinetic model in this stage is needed. In the final phase, strong electric fields, finite Larmor radius effect, high-energy (MeV) ion beam formation and beam-target fusion are present, thus kinetic model is also necessary to comprehend the clear picture in this process. Though there are many sophisticated existing code (MHD code: ALEGRA[16], FLASH[17], GORGON[18]; particle-in-cell (PIC) code: LSP[19], OSIRIS[20]), they are either controlled code or not appropriate for simulating DPF. A code suitable for the final pinch phase will be developed.

Highlights of the papers concerned with the numerical studies in pinch phase are as following:

- Kueny et al. in 2009[21] simulated the process of DPF by using a 2-D MHD code (ALEGRA) and the simulation results were validated by experimental results. First, they exported the data of the formation of the current sheaths from fully kinetic codes (LSP) without any assumptions of the structure of the current sheaths. ALE-

GRA was used to carry out the simulations from the run-down phases to the final pinch phase; the neutron yields and peak currents during the pinch phase was obtained at the end. The results showed that the neutron yields were underestimated as the authors expected since the MHD simulations could only predict thermonuclear component of neutrons. The neutrons produced by beam-target fusion were not considered in this case. Doing 3-D modeling in pinch phase for the next steps was concluded in this paper.

- Schmidt et al. in 2012[22] simulated the process of pinch phase in kJ DPF device using fully kinetic code (LSP) and the results reproduced the high-energy (MeV) ion beams and experimental neutron yields ($\sim 10^7$). The calculation was initialized by the assumption of the structure at the end of the run-down phase. Fluid and hybrid (kinetic ions and fluid electrons) models were also used for comparison. The results of fluid model was not allowed for nonthermal ions and showed no neutron yields; hybrid model underestimated the neutron yields and the energy of ions, while only fully kinetic model reproduced the experimental results. The authors believed that anomalous resistivity in the plasma in the pinch phase is attributed to lower hybrid drift instability according to the frequency analysis in fully kinetic simulations.

- Schmidt et al. in 2014[23] gave the first fully kinetic simulations of MJ DPF devices, which is much more computational expensive than that for kJ DPF devices due to the greater spatial scales involved. The simulations began with the plasma sheath at the insulator using 2D fluid model (ALEGRA). In run-in phase, the simulations were transferred into fully kinetic model using LSP. The results predicted D^+ energy distribution, neutron yields, anomalous resistivity and angular dependence of neutron production considering kinetic effects, such as beam-target fusion and ion beam formation. Moreover, neutron spectra from inside and outside of the pinch showed that thermonuclear fusion or low-energy beam-target fusion occurred outside of the pinch and beam-target fusion in the pinch.

- Link et al. in 2014[6] presented the hybrid fluid-to-kinetic model allowing the simu-

lations began in fluid mode from run-down phase to early part of run-in phase, and was transferred into fully kinetic model (LSP) in pinch phase. Simulations in the final pinch phase used fs and sub-mm particle scale to capture the kinetic instabilities containing anomalous resistivity, beam-target fusion and ion beam formation. The results showed that the different anode shapes would make a difference of the final neutron yields. Neutron anisotropy, neutron yield and ion beam formation were given in results of simulation, which were validated by empirical results.

- B. Appelbe and J. Chittenden in 2014[24] presented that they used 3D MHD model (GORGON) in run-down phases as well as a part of run-in phase and kinetic model in pinch phase to understand the neutron production occurring in DPF with 70 kA and 500 kA, respectively. The reason for the transition of fluid model to kinetic model was that strong electric fields combined with the low-density plasma in the final pinch phase. Anisotropic neutron emission, the important characteristics of beam-target fusion, were observed in their simulation results.

3 Future works

To do the simulation of pinch phase in DPF device, the procedures and progress of the project can be divided into coding and physical part. They are summarized in Table 1 and described in the following.

Step	*Coding Part*	*Physical Part*
1^{st}	Practicing modeling a simple project via PIC simulations.	Increasing my background knowledge of plasma physics.
2^{nd}	Optimizing my code by better numerical methods.	
3^{rd}	Developing PIC code for the final pinch phase of DPF.	Studying DPF and physics of z-pinch.
4^{th}	Accelerating the PIC code via computing parallelly using GPU.	
5^{th}	Building the plasma diagnostics for final pinch phase.	Studying mechanisms of neutron production via my simulation results and empirical results from INER

Table 1: Summary of the procedures and progress of the project

3.1 The 1^{st} and 2^{nd} steps

- Coding part: Since I have learned the numerical methods for solving PDE and ODE, I can directly write a PIC code for a simple project, e.g. Landau damping and two-stream instability, by reading Birdsall's textbook on plasma simulation[1] and some teaching notes online. After that, the optimization of my code through using better-performance numerical methods will be applied.

- Physical part: I only took one class related to plasma physics, space physics, which was taught by the teacher in Institute of space and plasma physics before. Increasing my background knowledge of plasma physics through taking classes and reading more textbooks or papers are necessary.

3.2 The 3^{rd} and 4^{th} steps

- Coding part: After writing a simple project, I can start developing PIC code for the final pinch phase of DPF. The initial condition of my simulations will be imported from the empirical condition of DPF device built by INER. If the simulation is too computational intensive, accelerating the PIC code via parallel computing using GPU will be needed.

- Physical part: Studying more papers about DPF and physics of z-pinch will be needed. A notable review article of Haines on z-pinch in 2011[25] will be a good material for understanding the physics of z-pinch.

3.3 The 5^{th} step

- Coding part: After finishing writing the PIC code for final pinch phase, I can get the results of neutron production, electric fields, ion densities and frequency analysis through plasma diagnostics. My simulation results will be validated using empirical results from INER and previous studies.

- Physical part: With the simulation results and experimental results, the mechanisms of the process of pinch phases e.g. neutron production, high-energy ion beam formation, beam-target fusion and anomalous resistivity etc. will be studied. Strong background knowledge of plasma physics related to z-pinch will be needed at this time.

4 Conclusion

Numerical methods for solving PDE were being practiced through developing the codes of numerical hydrodynamics, including solving ODE, parabolic PDE and linear advection equation. PIC simulations will be studied in this semester. As for the physical problem, DPF device is a pulsed-power device capable of producing 10^8 neutrons per shot and can be used as an efficient neutron source. However, the detailed mechanisms of neutron production in DPF are not fully understood. My research topic is to model the final pinch phase by developing a fully kinetic particle-in-cell code via computing parallelly using GPU from scratch and capable of predicting the neutron yields. Practicing modeling a simple physical problem like two-stream instability will be done in this semester.

References

[1] Charles K Birdsall and A Bruce Langdon. *Plasma physics via computer simulation*. CRC Press, 2004.

[2] Bernd Greytag. *Introduction to Numerical Hydrodynamics*. Naples, January 2010.

[3] Leopoldo Soto. New trends and future perspectives on plasma focus research. *Plasma Physics and Controlled Fusion*, 47(5A):A361, 2005.

[4] NV Filippov, TI Filippova, and VP Vinogradov. Dense high-temperature plasma in a non-cylindrical z-pinch compression. *Nucl. Fusion, Suppl.*, 1962.

[5] JW Mather. Investigation of the high-energy acceleration mode in the coaxial gun. *The Physics of Fluids*, 7(11):S28–S34, 1964.

[6] A Link, C Halvorson, EC Hagen, DV Rose, DR Welch, A Schmidt, Daniel Sinars, and Simon Bott-Suzuki. Particle-in-cell modeling for mj scale dense plasma focus with varied anode shape. In *AIP Conference Proceedings*, volume 1639, pages 23–26. AIP, 2014.

[7] Vincent Tang and Brian Rusnak. Review of dense plasma focus technology for intense and directional neutron sources. *Lawrence Livermore National Laboratory (LLNL-TR-401875),(February 29, 2008) pp*, pages 1–6, 2009.

[8] Mahadevan Krishnan. The dense plasma focus: A versatile dense pinch for diverse applications. *IEEE Transactions on Plasma Science*, 40(12):3189–3221, 2012.

[9] L Rapezzi, M Angelone, M Pillon, M Rapisarda, E Rossi, M Samuelli, and F Mezzetti. Development of a mobile and repetitive plasma focus. *Plasma Sources Science and Technology*, 13(2):272, 2004.

[10] Eric J Lerner, S Krupakar Murali, Derek Shannon, Aaron M Blake, and Fred Van Roessel. Fusion reactions from> 150 kev ions in a dense plasma focus plasmoid. *Physics of Plasmas*, 19(3):032704, 2012.

[11] Tim Warburton. Numerical methods for partial differential equations caam 452 - lecture 11, 2005.

[12] Tim Warburton. Numerical methods for partial differential equations caam 452 - lecture 12, 2005.

[13] John P Verboncoeur. Particle simulation of plasmas: review and advances. *Plasma Physics and Controlled Fusion*, 47(5A):A231, 2005.

[14] David Kirk et al. Nvidia cuda software and gpu parallel computing architecture. In *ISMM*, volume 7, pages 103–104, 2007.

[15] Xianglong Kong. *GPU accelerated particle-in-cell simulations with charge-conserving current deposition*. PhD thesis, Citeseer, 2013.

[16] Allen Robinson, Thomas Brunner, Susan Carroll, Richard Drake, Christopher Garasi, Thomas Gardiner, Thomas Haill, Heath Hanshaw, David Hensinger, Duane Labreche, et al. Alegra: An arbitrary lagrangian-eulerian multimaterial, multiphysics code. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, page 1235, 2008.

[17] A Dubey, LB Reid, and R Fisher. Introduction to flash 3.0, with application to supersonic turbulence. *Physica Scripta*, 2008(T132):014046, 2008.

[18] http://www.imperial.ac.uk/plasma-physics/magpie/numerical-simulations/.

[19] DR Welch, DV Rose, ME Cuneo, RB Campbell, and TA Mehlhorn. Integrated simulation of the generation and transport of proton beams from laser-target interaction. *Physics of Plasmas*, 13(6):063105, 2006.

[20] Ricardo A Fonseca, Luís O Silva, Frank S Tsung, Viktor K Decyk, Wei Lu, Chuang Ren, Warren B Mori, S Deng, S Lee, T Katsouleas, et al. Osiris: a three-dimensional, fully relativistic particle in cell code for modeling plasma based accelerators. In *International Conference on Computational Science*, pages 342–351. Springer, 2002.

[21] CS Kueny, DG Flicker, and DV Rose. Alegra-hedp simulations of the dense plasma focus. *Sandia National Laboratories, Albuquerque, NM, SAND2009-6373*, 2009.

[22] A Schmidt, V Tang, and D Welch. Fully kinetic simulations of dense plasma focus z-pinch devices. *Physical review letters*, 109(20):205003, 2012.

[23] A Schmidt, A Link, D Welch, BT Meehan, V Tang, C Halvorson, M May, and EC Hagen. Fully kinetic simulations of megajoule-scale dense plasma focus. *Physics of Plasmas*, 21(10):102703, 2014.

[24] Brian Appelbe, Jeremy Chittenden, Daniel Sinars, and Simon Bott-Suzuki. Understanding neutron production in the deuterium dense plasma focus. In *AIP Conference Proceedings*, volume 1639, pages 9–14. AIP, 2014.

[25] MG Haines. A review of the dense z-pinch. *Plasma Physics and Controlled Fusion*, 53(9):093001, 2011.