

National Cheng Kung University  
Institute of Space and Plasma Physics  
Annual Progress Report



1031  
研究生：楊宗桓

指導教授：張博宇 博士

中華民國一〇七年一月一日星期一

## 摘要

這份報告分為兩個部分：第一個部分是討論關於數值流體力學的模擬，在“Introduction to Numerical Hydrodynamics”這本書中一些用來解移流方程式的數值方法將會被驗證；第二個部分是在討論用來解釋無碰撞電漿時使用的伏拉索夫方程式 (Vlasov equation) 的數值方法，並被用來在模擬動力學理論下的電漿現象。



Abstract

There are two parts in this progress report. The first part is about the simulation verification of numerical hydrodynamics. The numerical methods in the book, “Introduction to Numerical Hydrodynamics” for solving advection equations were implemented for practicing numerical simulation. The second part is about the Vlasov solver. It is a fully kinetic model to calculate basic plasma phenomenon.



# Content

摘要 .....	2
Abstract.....	2
Content.....	4
Part I Numerical Hydrodynamics .....	6
Chapter 1 Practiced problems .....	6
1.1 Introduction .....	6
1.2 Advection equation.....	6
1.3 Discretization of advection equation .....	6
1.4 Initial conditions (ICs).....	9
1.5 Boundary conditions (BCs) .....	10
Chapter 2 Numerical results .....	10
2.1 Linear schemes for solving advection equation.....	10
2.1.1 Implicit centered scheme .....	11
Fig. 2-2 Numerical result of implicit centered scheme.....	12
2.1.2 Backward time center space (BTCS) scheme .....	12
Fig. 2-4 Numerical result of BTCS scheme.....	13
2.2 Nonlinear schemes for solving advection equation .....	13
2.2.1 Godunov-type finite volume scheme.....	13
2.2.2 The PLM schemes with linear slope-limiters .....	16
Fig. 2-6 Numerical result of FTBS scheme .....	17
2.2.3 PLM schemes with nonlinear slope-limiters .....	22
2.2.4 Piecewise parabolic reconstruction (PPM) method .....	27
Part II Vlasov solver.....	30

Chapter 3 Introduction to Vlasov solver .....	30
3.1 Basic equations of Vlasov solver.....	30
3.2 Normalization of basic equations .....	31
3.3 Simulation order of Vlasov solver .....	33
3.4 Numerical structure .....	33
3.5 Numerical space grids .....	34
Chapter 4 Numerical methods and simulation results .....	35
4.1 Initial condition.....	36
4.2 Boundary condition.....	37
4.3 Split Vlasov equations .....	38
4.3.1 Split advection equation in x direction .....	39
4.3.2 Split advection equation in v direction .....	40
4.4 Charge density.....	41
4.5 Poisson's equation .....	42
4.4 Electric field and acceleration.....	45
Chapter 5 Future work .....	47
5.1 Short term goal .....	47
5.2 Long term goal .....	47
Chapter 6 Summary .....	48
Reference.....	49

# Part I Numerical Hydrodynamics

## Chapter 1 Practiced problems

### 1.1 Introduction

We would like to study the behaviors of plasma in plasma in kinetic regime numerically. Some complex equations will be solved. These equations can be solved numerically and different numerical schemes are studied by following the textbook, "Introduction to Numerical Hydrodynamics" [1]. Details will be given in the following sections.

### 1.2 Advection equation

The advection equations have been simulated in different numerical schemes. The advection equation belongs to hyperbolic equation. It is used to express the advection flow. It is written as

$$\frac{\partial \rho}{\partial t} = v \frac{\partial \rho}{\partial x}. \quad (1-1)$$

where  $\rho = \rho(x, t)$  is fluid density,  $v$  is velocity,  $x$  is space, and  $t$  is time. Note that both  $\rho$  and  $v$  are functions of  $x$  and  $t$ . The analytical solution of this advection equation should be  $\rho(x, t) = \rho(x - vt)$ .

### 1.3 Discretization of advection equation

To solve the advection equation numerically, partial differentiates in Eq.(1-1) can be written in the differentiate forms as

$$\frac{\partial \rho}{\partial t} \rightarrow \frac{\rho_i^{n+1} - \rho_i^n}{\delta t},$$

$$v \frac{\partial \rho}{\partial x} \rightarrow v \frac{\rho_{i+1}^n - \rho_{i-1}^n}{2\delta x}.$$

Time and space also can be discretized as

$$t^n = \delta t * n + t_0,$$

$$x_i = \delta x * i + x_0.$$

where  $\delta x = \frac{1}{i_{\max}}$ , and  $\delta t = \frac{1}{t_{\max}}$ . The total time we solved here is 1. As figure 1-1 shows in below, the subscript “n” refers to the quantities in the time step  $t_n$ , and the subscript “i” refers to the discretized location  $x_i$ . When we solved the equation, we had  $i_{\max} = 200$ , and  $t_{\max} = 500$ , so that  $\delta x = 0.005$ , and  $\delta t = 0.002$ . To achieve the numerical stable, Courant-Friedrichs-Levy (CFL) condition defines as  $\frac{\delta t}{\delta x} v < 1$  is required. For  $v = 1$  in examples in the next chapter,  $\frac{\delta t}{\delta x} v = 0.4$ , which satisfy the CFL condition.

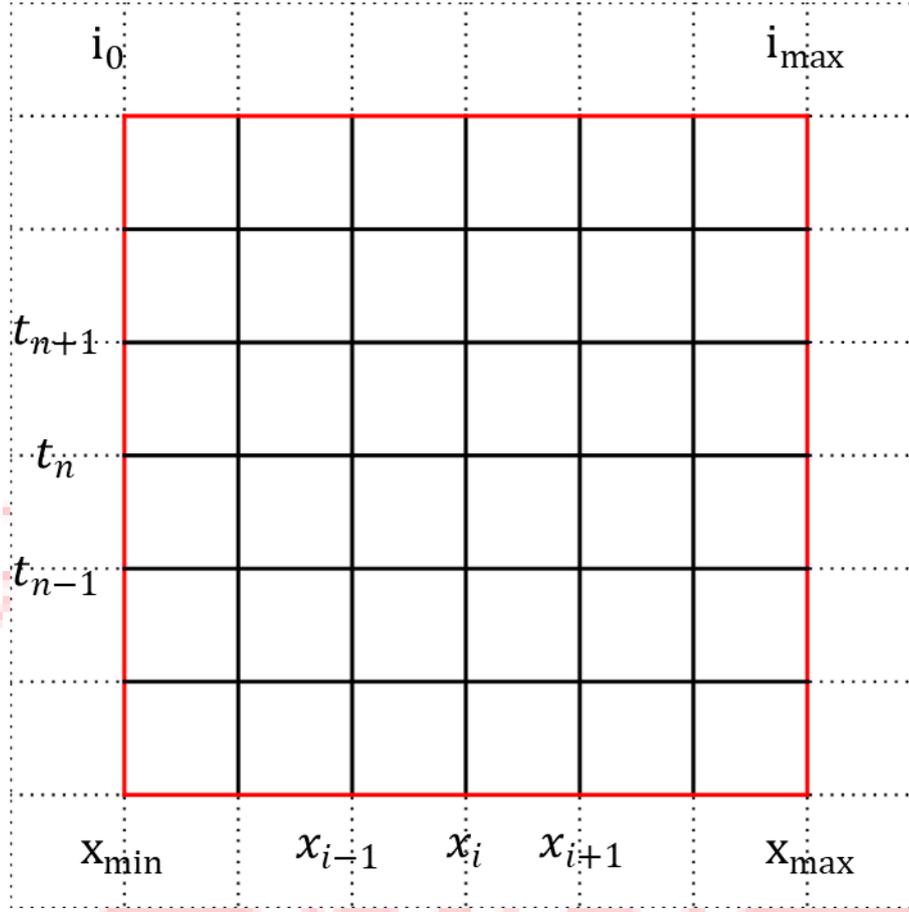


Fig.1-1 Simulation grids

As a result, Eq.(1-1) can be written as Eq.(1-2).

$$\frac{\rho_i^{n+1} - \rho_i^n}{\delta t} = v \frac{\rho_{i+1}^n - \rho_{i-1}^n}{2\delta x}. \quad (1-2)$$

Finally, the density in the new time step  $\rho_i^{n+1}$  can be calculated by the one using at the old time step, so Eq.(1-2) can be changed as Eq.(1-3) as below.

$$\rho_i^{n+1} = \rho_i^n - \frac{\delta t}{2\delta x} v (\rho_{i+1}^n - \rho_{i-1}^n). \quad (1-3)$$

The above equation can be rewritten using the flux of fluid flow.  $f_i^n$  defined as  $f_i^n = f(\rho_i^n) = v\rho_{i+1}^n$ . As a result, Eq.(1-3) is written as Eq.(1-4).

$$\rho_i^{n+1} = \rho_i^n - \frac{\delta t}{2\delta x} (f_{i+1}^n - f_{i-1}^n). \quad (1-4)$$

Furthermore,  $f_{i+1}^n$  and  $f_{i-1}^n$  are replaced by  $f_{i+\frac{1}{2}}^n$  and  $f_{i-\frac{1}{2}}^n$  where  $f_{i+\frac{1}{2}}^n = \frac{f_{i+1}^n + f_i^n}{2}$  and  $f_{i-\frac{1}{2}}^n = \frac{f_{i-1}^n + f_i^n}{2}$  leading to

$$\rho_i^{n+1} = \rho_i^n - \frac{\delta t}{\delta x} (f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n). \quad (1-5)$$

## 1.4 Initial conditions (ICs)

To practice different numerical schemes, we used the following initial conditions:

(1) A Gaussian function with center at  $x = 0.15$ : at  $0.1 < x < 0.2$ ,  $f(x) = e^{-\frac{(x-0.15)^2}{2\sigma^2}}$ ,  $\sigma=0.01$ .

(2) A rectangle function: at  $0.3 < x < 0.4$ ,  $f(x) = 1$ .

(3) A triangle function with peak at  $x = 0.55$ : at  $0.5 < x < 0.55$ ,  $f(x) = 20x - 10$ , and at  $0.55 < x < 0.6$ ,  $f(x) = 12 - 20x$ .

(4) A half-ellipse function with peak at  $x = 0.75$ : at  $0.7 < x < 0.8$ ,  $f(x) = b\sqrt{1 - \frac{(x-0.75)^2}{a^2}}$ ,  $b=1$ ,  $a=0.75$ .

(5) Otherwise,  $f(x) = 0$ .

The initial condition is shown in figure 1-2.

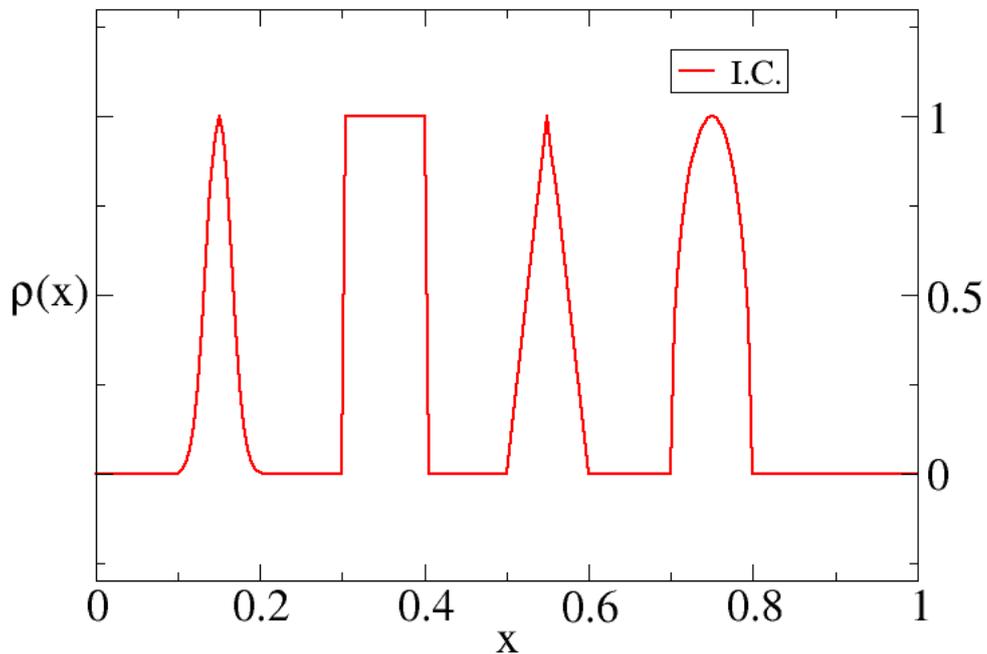


Fig.1-2 Initial condition

### 1.5 Boundary conditions (BCs)

Periodic boundary conditions are used here. A right-advected wave shows up from the left boundary when it vanished from the right boundary. The real boundary is set as  $\rho(0, t) = \rho(x_{\max}, t)$ ,  $\rho(x_{\max}, t) = \rho(0, t)$ , so the boundaries can be set as  $\rho_{i_{\max}+1} = \rho_{i_{\min}+1}$ , and  $\rho_{i_{\min}-1} = \rho_{i_{\max}-1}$  numerically.

## Chapter 2 Numerical results

Numerical results of the advection equations using some numerical methods are given in this chapter. They can be categorized as linear schemes and nonlinear schemes separately.

### 2.1 Linear schemes for solving advection equation

In this section, linear numerical schemes are used. The parameters of advection equation were given in section 1.3, and the initial conditions and boundary conditions were given in section 1.4 and section 1.5 respectively

### 2.1.1 Implicit centered scheme

The flux of implicit centered scheme in Eq.(1-5) is

$$f_{i+\frac{1}{2}}^n = \frac{1}{4}(v\rho_{i+1}^n + v\rho_i^n) + \frac{1}{4}(v\rho_{i+1}^{n+1} + v\rho_i^{n+1}),$$

$$f_{i-\frac{1}{2}}^n = \frac{1}{4}(v\rho_i^n + v\rho_{i-1}^n) + \frac{1}{4}(v\rho_i^{n+1} + v\rho_{i-1}^{n+1}).$$

The stencil diagram of implicit centered scheme is shown in figure 2-1.

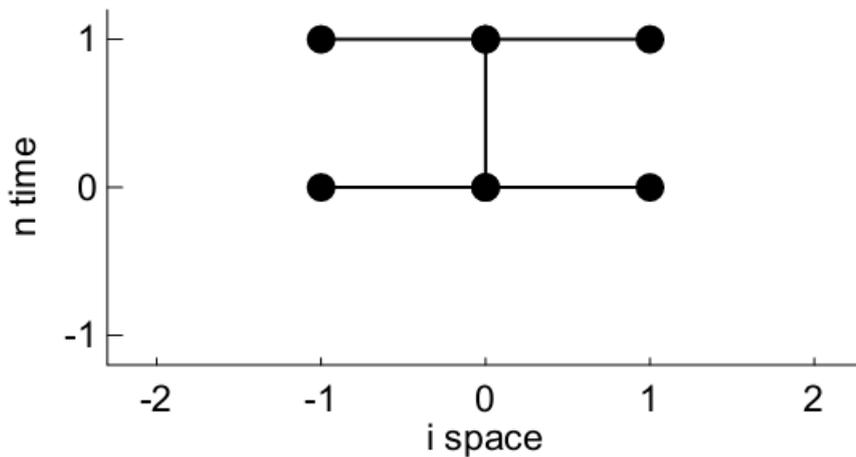


Fig.2-1 Stencil diagram of implicit centered scheme

The numerical result of implicit centered scheme is shown in figure 2-2

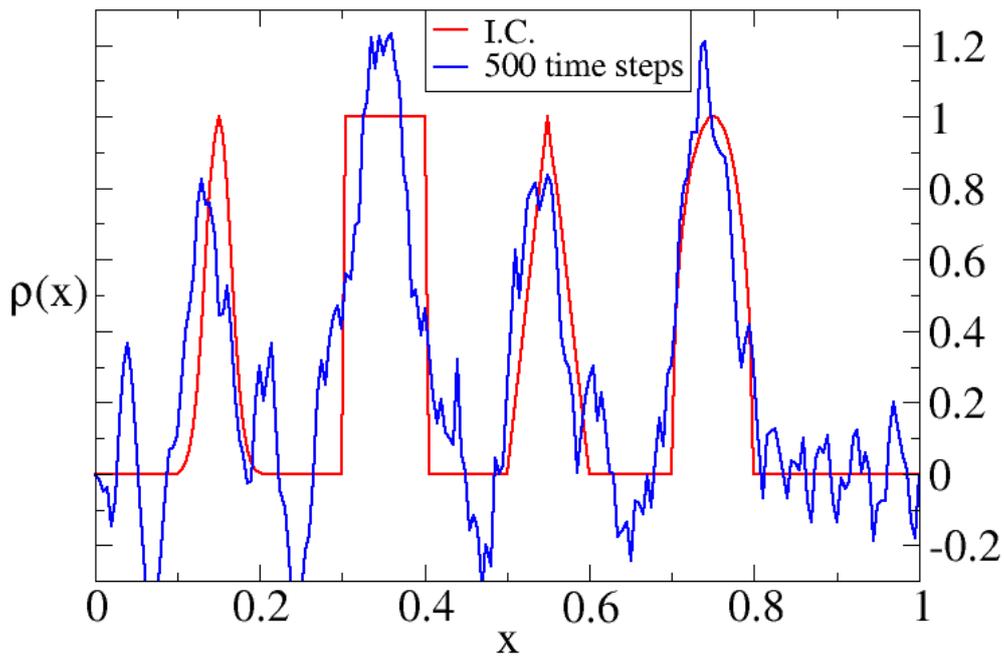


Fig. 2-2 Numerical result of implicit centered scheme

Strong numerical oscillations are observed in the simulation result using implicit centered scheme.

### 2.1.2 Backward time center space (BTCS) scheme

The flux of backward time center space in Eq.(1-5) is

$$f_{i+\frac{1}{2}}^n = \frac{1}{2}(v\rho_{i+1}^{n+1} + v\rho_i^{n+1}),$$

$$f_{i-\frac{1}{2}}^n = \frac{1}{2}(v\rho_i^{n+1} + v\rho_{i-1}^{n+1}).$$

The stencil diagram of BTCS scheme is shown in figure 2-3.

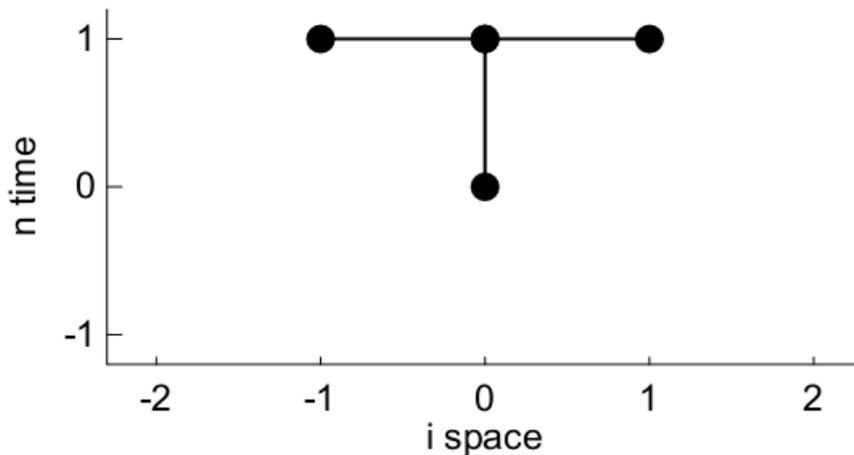


Fig. 2-3 Stencil diagram of BTCS scheme

The numerical result of BTCS scheme is shown in figure 2-4.

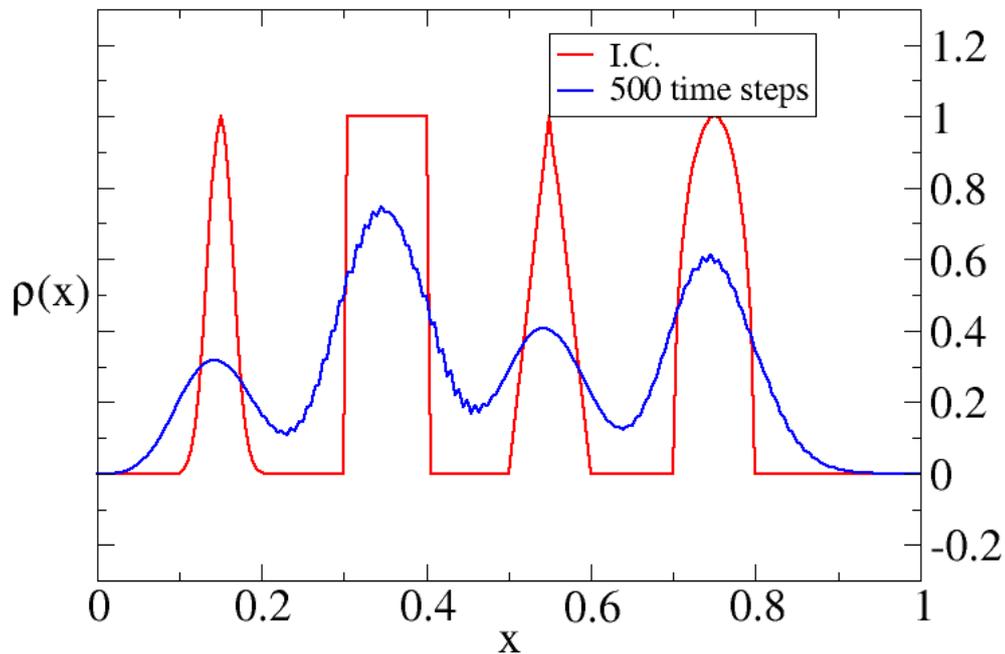


Fig. 2-4 Numerical result of BTCS scheme

Small zig-zag wiggles and diffusion phenomena are observed after propagating.

## 2.2 Nonlinear schemes for solving advection equation

In this section, nonlinear schemes are used to solve the advection equation. The parameters of advection equation are given in section 1.3, and the initial conditions and boundary conditions were given in section 1.4 and section 1.5 respectively.

### 2.2.1 Godunov-type finite volume scheme

The Godunov's idea is a kind of discretized method to solve nonlinear partial differential equations. It can be separated to three steps which are called Reconstruct-Solve-Average (RSA).

(1) "Reconstruct" a continuous function  $\rho(x)$  to a discrete function  $\rho_i$ .

(2) "Solve" for  $\delta t$ .

(3) "Average  $\rho(x)$ " in each cell to get  $\rho_i$ .

After using finite volume method to solve Eq.(1-5) between  $x_{i+\frac{1}{2}}$  and  $x_{i-\frac{1}{2}}$  in space and  $t^{n+1}$  and  $t^n$  in time, Eq.(1-5) becomes the following form.

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (\rho_i^{n+1} - \rho_i^n) dx = \int_{t^n}^{t^{n+1}} (f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n) dt. \quad (2.1)$$

If we defined the average of  $\rho_i^n$  in space grids,  $\bar{\rho}_i^{n+1} - \bar{\rho}_i^n$  is  $\frac{\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (\rho_i^{n+1} - \rho_i^n) dx}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}$ ,

where  $x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} = \delta x$ . Then Eq.(2.1) could be changed to

$$\begin{aligned} \bar{\rho}_i^{n+1} &= \bar{\rho}_i^n - \frac{1}{\delta x} \int_{t^n}^{t^{n+1}} (f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n) dt \\ &= \bar{\rho}_i^n - \frac{v}{\delta x} \int_{t^n}^{t^{n+1}} \left( \rho \left( x_{i+\frac{1}{2}}, t^n \right) - \rho \left( x_{i-\frac{1}{2}}, t^n \right) \right) dt. \end{aligned} \quad (2.2)$$

The piecewise linear method (PLM) is used to reconstruct linear function in each cell, which means the function  $\rho(x, t)$  can be approximated with 1<sup>st</sup>-order Taylor's expansion. The  $\rho(x)$  can be written as

$$\rho(x_i, t^n) = \rho_i^n + (x - x_i) \delta \rho_i^n. \quad (2.3)$$

where slope-limiter  $\delta \rho_i^n$  is the average of grids which is defined as  $\frac{d\rho}{dx}$ . If we put

Eq.(2.3) into the finite volume equation Eq.(2.2), and calculate the integral with

$x = x_{i+\frac{1}{2}} + v(t - t^n)$  and  $x = x_{i-\frac{1}{2}} + v(t - t^n)$  when  $x$  approximates to  $x_{i+\frac{1}{2}}$  and  $x_{i-\frac{1}{2}}$

respectively. Then, Eq.(2.2) can become

$$\bar{\rho}_i^{n+1} = \bar{\rho}_i^n - \frac{\delta t}{\delta x} (f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n). \quad (2.4)$$

where the  $f_{i+\frac{1}{2}}^n$  and  $f_{i-\frac{1}{2}}^n$  can be known as

$$f_{i+\frac{1}{2}}^n = v[\rho_i^n + \frac{1}{2}\delta\rho_i^n \text{sign}(v) (1 - |v| \frac{\delta t}{\delta x})],$$

$$f_{i-\frac{1}{2}}^n = v[\rho_{i-1}^n + \frac{1}{2}\delta\rho_{i-1}^n \text{sign}(v) (1 - |v| \frac{\delta t}{\delta x})].$$

The  $\text{sign}(v) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases}$  is set as a sign function.

Finally,  $f_{i+\frac{1}{2}}^n$  and  $f_{i-\frac{1}{2}}^n$  terms are substituted into the finite difference Eq.(2-4).

$$\rho_i^{n+1} = \rho_i^n - \frac{\delta t}{\delta x} v \left[ (\rho_i^n - \rho_{i-1}^n) + \frac{1}{2} (\delta\rho_i^n - \delta\rho_{i-1}^n) \text{sign}(v) (1 - |v| \frac{\delta t}{\delta x}) \right]. \quad (2.5)$$

$f_{i+\frac{1}{2}}^n$  and  $f_{i-\frac{1}{2}}^n$  also can be simplified as following form by ignoring the boundary grids in space, which means  $|v| \frac{\delta t}{\delta x}$  term.

$$f_{i+\frac{1}{2}}^n = v[\rho_i^n + \frac{1}{2}\delta\rho_i^n \text{sign}(v)],$$

$$f_{i-\frac{1}{2}}^n = v[\rho_{i-1}^n + \frac{1}{2}\delta\rho_{i-1}^n \text{sign}(v)].$$

The advection equation can be simplified to the form.

$$\rho_i^{n+1} = \rho_i^n - \frac{\delta t}{\delta x} v \left[ (\rho_i^n - \rho_{i-1}^n) + \frac{1}{2} (\delta\rho_i^n - \delta\rho_{i-1}^n) \text{sign}(v) \right].$$

## 2.2.2 The PLM schemes with linear slope-limiters

There are some methods with linear slope-limiter are shown in this sections.

### Donor cell (Forward time backward space, FTBS) scheme:

The slope-limiter of FTBS scheme is

$$\delta\rho_i^n = 0.$$

Eq.(1-5) becomes

$$\rho_i^{n+1} = \rho_i^n - \frac{\partial t}{\partial x} v(f_i^n - f_{i-1}^n).$$

The stencil diagram of FTBS scheme is shown in figure 2-5.

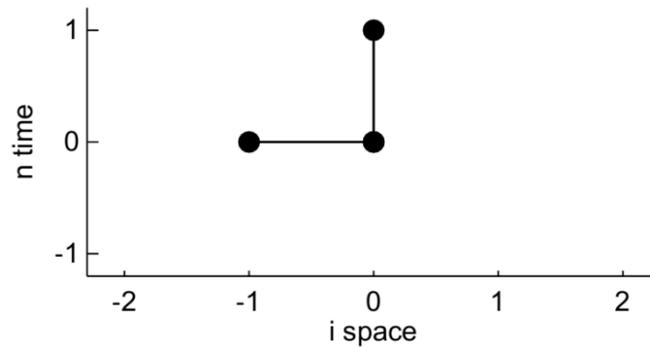


Fig. 2-5 Stencil diagram of FTBS scheme

The numerical result of FTBS scheme is shown in figure 2-6.

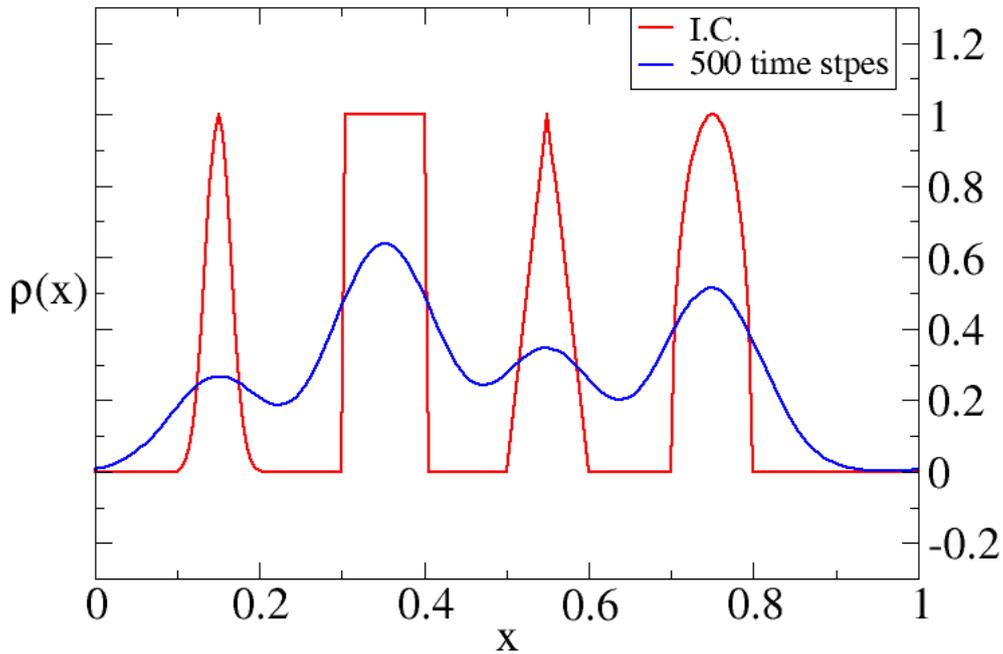


Fig. 2-6 Numerical result of FTBS scheme

Strong diffusion phenomena are observed after propagating.

### Lax-Wendroff scheme:

The slope-limiter of Lax-Wendroff scheme is

$$\delta \rho_i^n = \rho_{i+1}^n - \rho_i^n.$$

Eq.(1-5) becomes

$$\rho_i^{n+1} = \rho_i^n - \frac{\partial t}{\partial x} v (f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n),$$

$$f_{i+\frac{1}{2}}^n = \frac{1}{2} (f_{i+1}^n + f_i^n) - \frac{\partial t}{2 \partial x} v (f_{i+1}^n - f_i^n),$$

$$f_{i-\frac{1}{2}}^n = \frac{1}{2} (f_i^n + f_{i-1}^n) - \frac{\partial t}{2 \partial x} v (f_i^n - f_{i-1}^n).$$

The stencil diagram of Lax-Wendroff scheme is shown in figure 2-7.

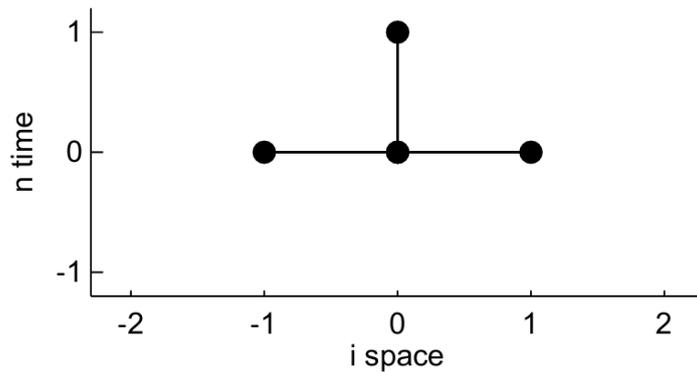


Fig. 2-7 Stencil diagram of Lax-Wendroff scheme

The numerical result of Lax-Wendroff scheme is shown in figure 2-8

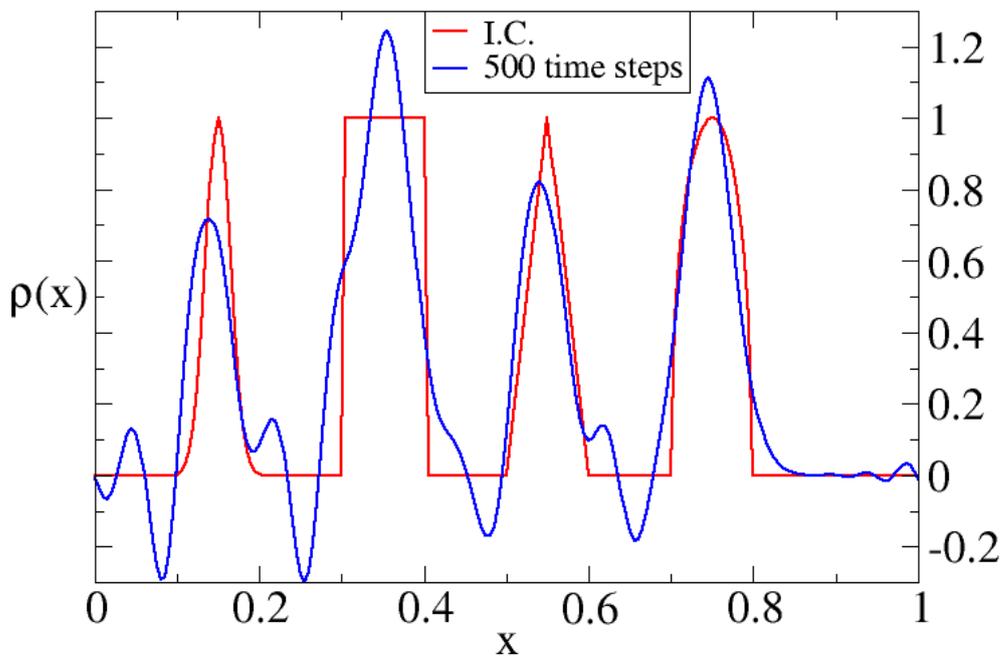


Fig. 2-8 Numerical result of Lax-Wendroff scheme

Strong oscillations are observed after density profiles propagate.

**Beam-Warming scheme:**

The slope-limiter of Beam-Warming scheme is

$$\delta\rho_i^n = \rho_i^n - \rho_{i-1}^n.$$

Eq.(1-5) becomes

$$\rho_i^{n+1} = \rho_i^n - \frac{\partial t}{\partial x} v(f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n),$$

$$f_{i+\frac{1}{2}}^n = \frac{1}{2}(3f_i^n - f_{i-1}^n) - \frac{\partial t}{2\partial x} v(f_i^n - f_{i-1}^n),$$

$$f_{i-\frac{1}{2}}^n = \frac{1}{2}(3f_{i-1}^n - f_{i-2}^n) - \frac{\partial t}{2\partial x} v(f_{i-1}^n - f_{i-2}^n).$$

The stencil diagram of Beam-Warming scheme is shown in figure 2-9.

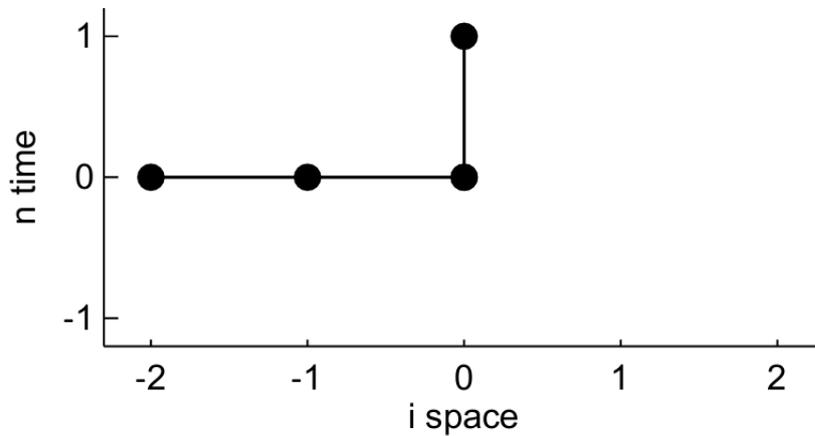


Fig. 2-9 Stencil diagram of Beam-Warming scheme

The numerical result of Beam-Warming scheme is shown in figure 2-10

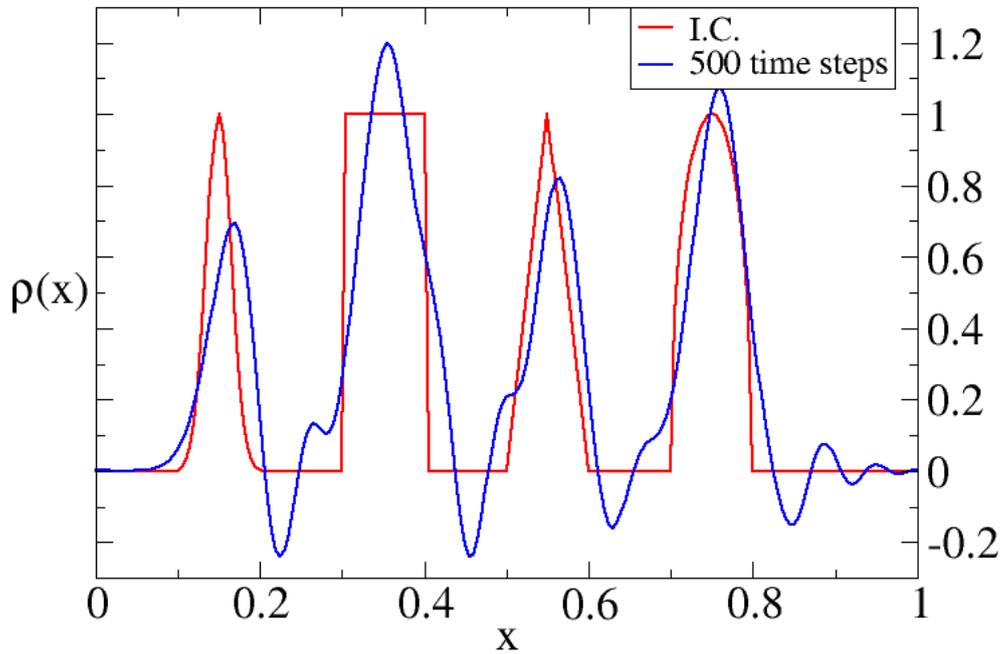


Fig. 2-10 Numerical result of Beam-Warming scheme

Strong oscillations are observed after propagating.

**Fromm scheme:**

The slope-limiter of Lax-Wendroff scheme is

$$\delta\rho_i^n = \frac{1}{2}(\rho_{i+1}^n - \rho_{i-1}^n).$$

Eq.(1-5) becomes

$$\rho_i^{n+1} = \rho_i^n - \frac{\partial t}{\partial x} v(f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n),$$

$$f_{i+\frac{1}{2}}^n = f_{i+\frac{1}{2}}^n(\text{Lax - Wendroff}) + f_{i+\frac{1}{2}}^n(\text{Beam - Warming}),$$

$$f_{i-\frac{1}{2}}^n = f_{i-\frac{1}{2}}^n(\text{Lax - Wendroff}) + f_{i-\frac{1}{2}}^n(\text{Beam - Warming}).$$

The stencil diagram of Fromm scheme is shown in figure 2-11.

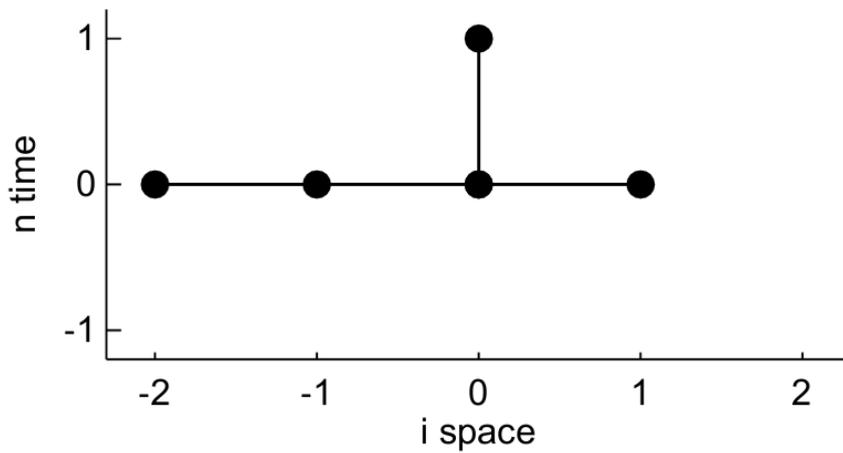


Fig. 2-11 Stencil diagram of Fromm scheme

The numerical result of Fromm scheme is shown in figure 2-12

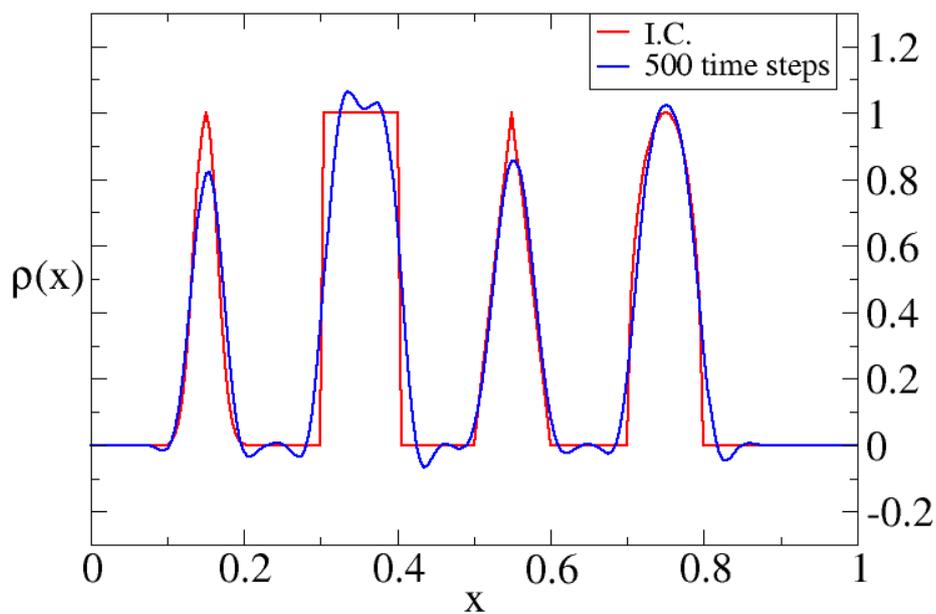


Fig. 2-12 Numerical result of Fromm scheme

The scheme is observed almost the same graph as initial state with some instable at peaks.

### 2.2.3 PLM schemes with nonlinear slope-limiters

All slope-limiters in this section use the same stencil diagram in figure 2-13.

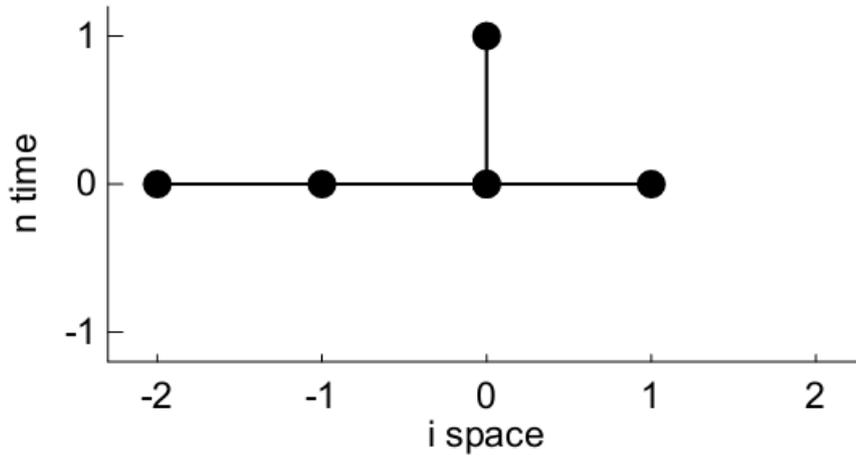


Fig. 2-13 Stencil diagram of PLM scheme

#### **PLM scheme with Minmod slope limiter:**

Minmod slope-limiter is defined as

$$\delta\rho_i^n = \min(\max(\rho_i^n - \rho_{i-1}^n, 0), \max(\rho_{i+1}^n - \rho_i^n, 0)) + \max(\min(\rho_i^n - \rho_{i-1}^n, 0), \min(\rho_{i+1}^n - \rho_i^n, 0)).$$

The numerical result of Minmod slope-limiter is shown in figure 2-14.

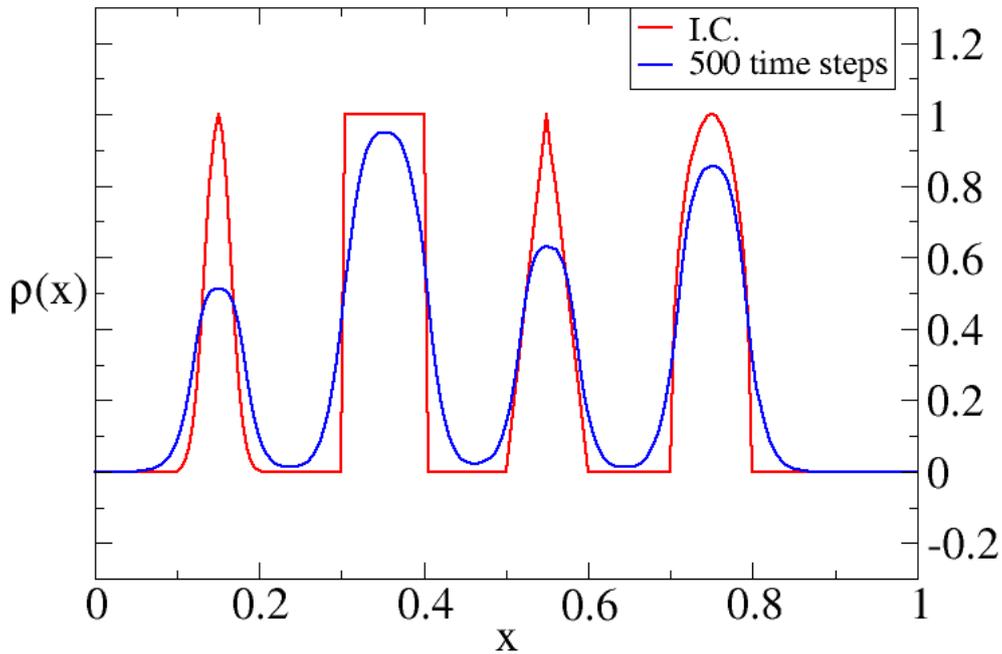


Fig. 2-14 Numerical result of Minmod slope-limiter

Strong diffusion is observed at peaks.

**PLM scheme with vanLeer slope-limiter:**

The vanLeer slope-limiter is defined as

$$\delta\rho_i^n = \begin{cases} 2 \frac{(\rho_{i+1}^n - \rho_i^n)(\rho_i^n - \rho_{i-1}^n)}{\rho_{i+1}^n - \rho_{i-1}^n}, & \text{if } (\rho_{i+1}^n - \rho_i^n)(\rho_i^n - \rho_{i-1}^n) > 0 \\ 0, & \text{others} \end{cases}$$

The numerical result of vanLeer slope-limiter is shown in figure 2-15.

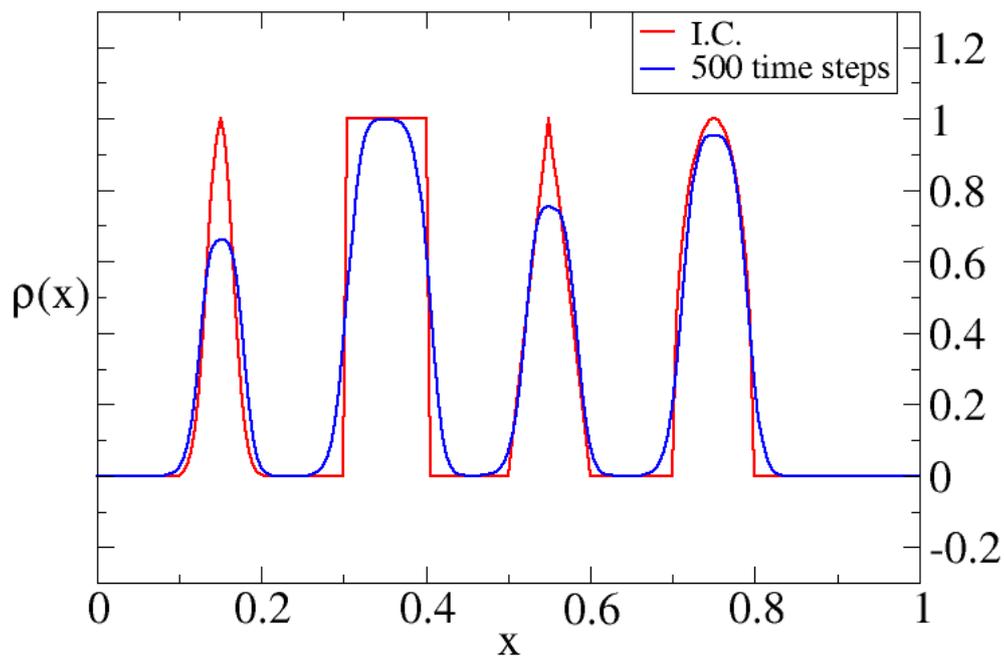


Fig. 2-15 Numerical result of vanLeer

Strong diffusion is observed at peaks, but weaker than Minmod one.

#### **PLM scheme with Superbee slope-limiter:**

The Superbee slope-limiter is defined as

$$\delta\rho_i^n = [\text{sign}(\rho_i^n - \rho_{i-1}^n) + \text{sign}(\rho_{i+1}^n - \rho_i^n)] \times \min[|\rho_i^n - \rho_{i-1}^n|, |\rho_{i+1}^n - \rho_i^n|, \frac{1}{2} \max(|\rho_i^n - \rho_{i-1}^n|, |\rho_{i+1}^n - \rho_i^n|)].$$

The numerical result of Superbee slope-limiter is shown in figure 2-16.

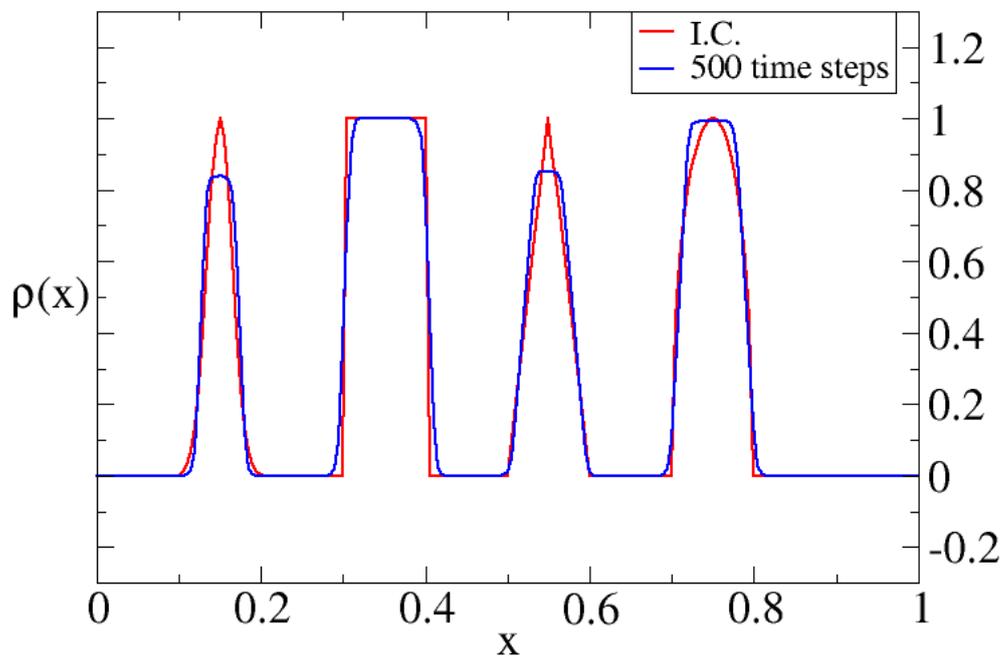


Fig. 2-16 Numerical result of Superbee slope-limiter

A little diffusion is observed at peaks, but weaker than both vanLeer and Minmod one. Some expansions are also observed at peaks.

### Modified PLM scheme with Superbee slope-limiter:

This method uses simplified PLM scheme equation with Superbee slope-limiter.

The numerical result of Superbee slope-limiter is shown in figure 2-17.

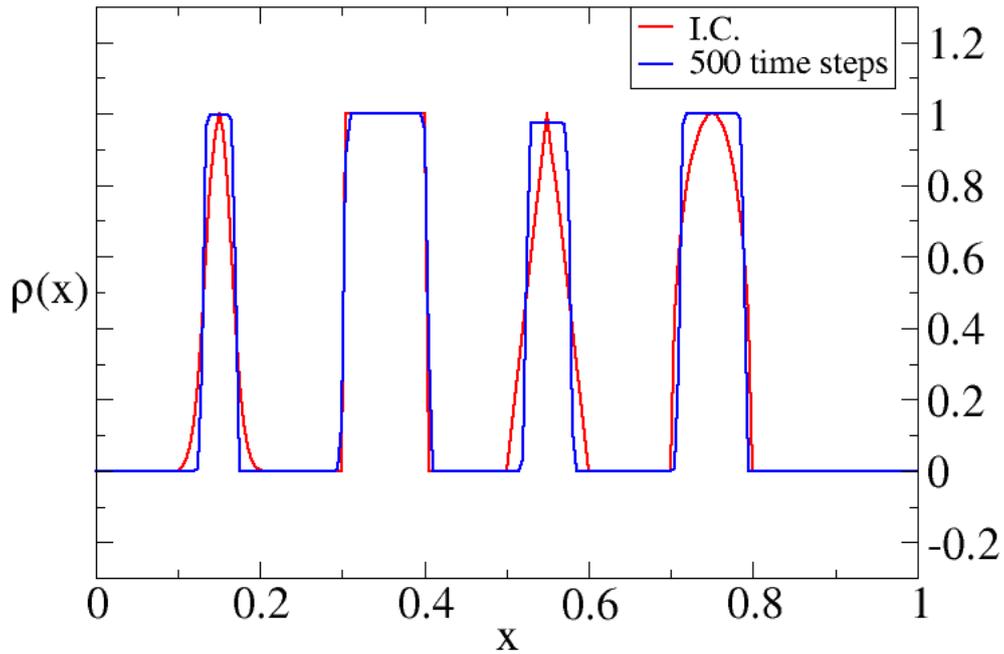


Fig. 2-17 Numerical result of Modified Superbee slope-limiter

Some expansions are observed at peaks in this scheme, but it still shows a great simulation result.

## 2.2.4 Piecewise parabolic reconstruction (PPM) method

PPM method is a finite volume method also used Godunov's scheme but with higher-order approximations which is a 2<sup>nd</sup>-order parabolic reconstruction. As a result, the Eq.(2.3) can be replaced by a Taylor's expansion which is a parabolic approximation as

$$\rho(x_i, t^n) = \rho_i^n + (x - x_i)\delta\rho_i^n + (x - x_i)^2\delta^2\rho_i^n$$

This equation can be brought into Eq.(2.2) as above and become the form as Eq.(2.4).

$$\bar{\rho}_i^{n+1} = \bar{\rho}_i^n - \frac{\delta t}{\delta x} (f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n). \quad (2.4)$$

As in Eq.(2-4), the  $f_{i+\frac{1}{2}}^n$  and  $f_{i-\frac{1}{2}}^n$  of PPM are set as

$$f_{i+\frac{1}{2}}^n = v\rho_{i+\frac{1}{2}}^n - \frac{v^2\delta t}{2\delta x} \left[ \left( \rho_{i+\frac{1}{2}}^n - \rho_{i-\frac{1}{2}}^n \right) - \left( 1 - \frac{2v\delta t}{3\delta x} \rho_i^n \right) \right],$$

$$f_{i-\frac{1}{2}}^n = v\rho_{i-\frac{1}{2}}^n - \frac{v^2\delta t}{2\delta x} \left[ \left( \rho_{i-\frac{1}{2}}^n - \rho_{i-\frac{3}{2}}^n \right) - \left( 1 - \frac{2v\delta t}{3\delta x} \rho_{i-1}^n \right) \right].$$

The parameters in the  $f_{i+\frac{1}{2}}^n$  and  $f_{i-\frac{1}{2}}^n$  are set as

$$\rho_{i+\frac{1}{2}}^n = \left[ \frac{7}{12} (\rho_i^n + \rho_{i+1}^n) - \frac{1}{12} (\rho_{i+2}^n + \rho_{i-1}^n) \right]$$

$$\rho_{i-\frac{1}{2}}^n = \left[ \frac{7}{12} (\rho_{i-1}^n + \rho_i^n) - \frac{1}{12} (\rho_{i+1}^n + \rho_{i-2}^n) \right]$$

$$\rho_{i-\frac{3}{2}}^n = \left[ \frac{7}{12} (\rho_{i-2}^n + \rho_{i-1}^n) - \frac{1}{12} (\rho_i^n + \rho_{i-3}^n) \right]$$

$$\rho_i^n = \left[ 6\rho_i^n - \frac{1}{2} (\rho_{i+\frac{1}{2}}^n + \rho_{i-\frac{1}{2}}^n) \right]$$

$$\rho_{i-1}^n = \left[ 6\rho_{i-1}^n - \frac{1}{2} (\rho_{i-\frac{1}{2}}^n + \rho_{i-\frac{3}{2}}^n) \right]$$

The stencil diagram of PPM scheme is shown in figure 2-18.

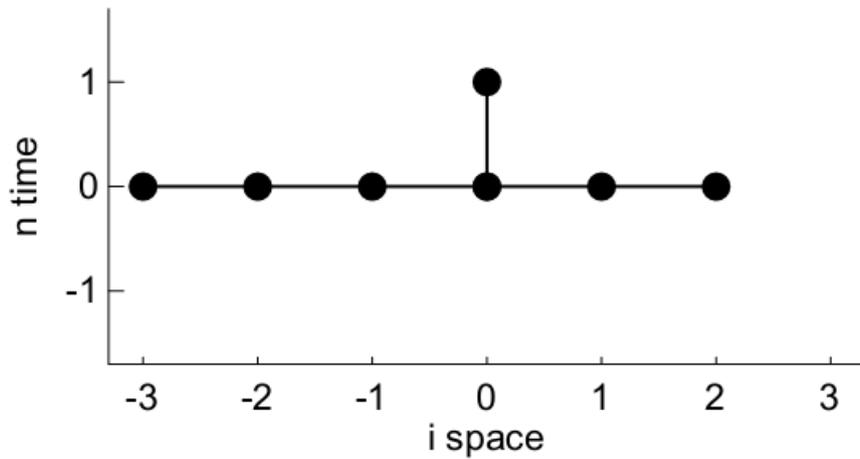


Fig. 2-18 Stencil diagram of PPM scheme

The numerical result of PPM scheme is shown in figure 2-19.

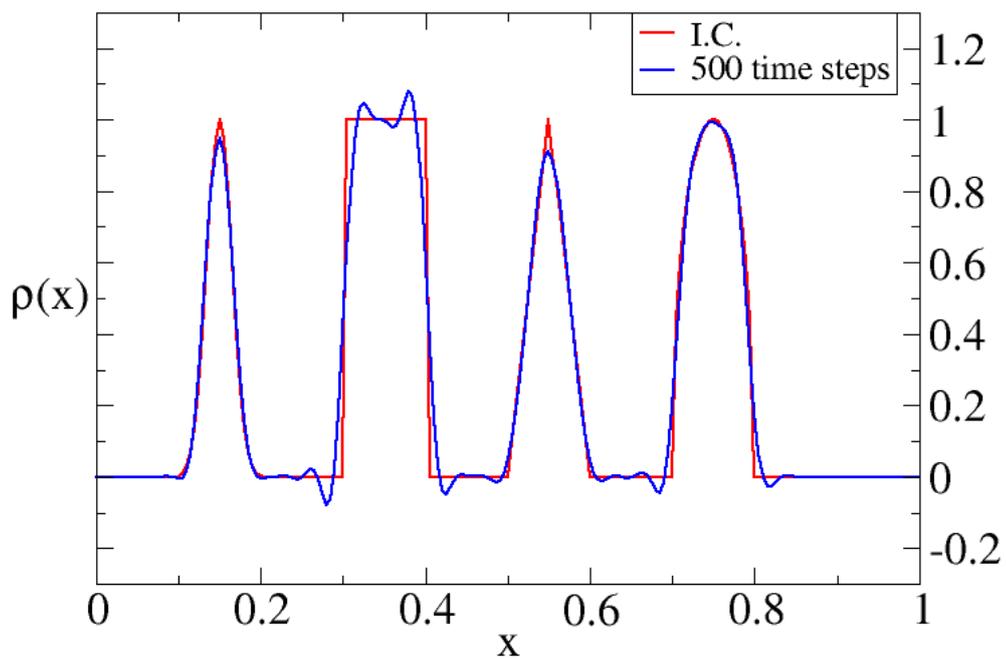


Fig. 2-19 Numerical result of PPM scheme

Some zigzags are observed at bottoms and peaks, but PPM scheme shows the best simulation result in all simulation. The PLM scheme with Superbee slope-limiter is the second best one followed by the PLM scheme with VanLeer slope-limiter., The PLM

scheme with Minmod slope-limiter is the worst used nonlinear method. In linear method, the Fromm scheme shows the best perform as PPM scheme, but the other methods can't be used because of huge oscillations or diffusions.



## Part II Vlasov solver

### Chapter 3 Introduction to Vlasov solver

#### 3.1 Basic equations of Vlasov solver

Vlasov equation is used to calculate plasma phenomena without collision in kinetic theory. The Vlasov solver is a numerical method to simulate plasma phenomena by solving Vlasov equation directly. It shows high precision compared to other methods but with huge calculation time.

One-dimensional (1-D) Vlasov-Poisson system is used. The main equations in this system have the form.

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} + a \frac{\partial f}{\partial v} = 0, \text{ Vlasov equation.} \quad (3-1)$$

$$\frac{\partial^2 \varphi}{\partial x^2} = -\frac{e}{\epsilon_0} (n_i - \int f dv), \text{ Poisson's equation.} \quad (3-2)$$

In Vlasov equation as Eq.(3-1), the distribution function  $f(x,v,t)$  is generally given as an initial condition,  $v$  is velocity and  $a$  is acceleration.  $\varphi(x)$  is electric potential here in Eq.(3-2), and  $q$  is electric quantity.  $n_i$  on the right-hand side means the number density of ions and is set as 1, and  $\int f dv$  is the number density of electrons. To calculate the acceleration  $a$ , we need to use the Poisson's equation as Eq.(3-2). Then the electric potential  $\varphi$  has the relationship with acceleration  $a$  as

$$E = -\frac{d\varphi}{dx}. \quad (3-3)$$

$$a = -\frac{eE}{m_e} \quad (3-4)$$

### 3.2 Normalization of basic equations

To simulate the equations above easily without complex constants, we have to normalize these equations to dimensionless equations to ensure this simulation is correct. There are some characteristic units used to normalize equations here.

$$t = t_{\text{char}} \hat{t}.$$

$$x = x_{\text{char}} \hat{x}.$$

$$n = n_{\text{char}} \hat{n}.$$

$$v = v_{\text{char}} \hat{v}.$$

$$f = f_{\text{char}} \hat{f}.$$

$$a = a_{\text{char}} \hat{a}.$$

$$\varphi = \varphi_{\text{char}} \hat{\varphi}.$$

$$E = E_{\text{char}} \hat{E}.$$

The characteristic units above are set as below.

$$t_{\text{char}} = \omega_p^{-1} \quad \omega_p \text{ is the plasma frequency.}$$

$$x_{\text{char}} = v_{\text{char}} t_{\text{char}}$$

$$n_{\text{char}} = f_{\text{char}} v_{\text{char}}$$

$$\varphi_{\text{char}} = \frac{q f_{\text{char}} v_{\text{char}} x_{\text{char}}^2}{\epsilon_0}$$

$$E_{\text{char}} = \frac{\varphi_{\text{char}}}{x_{\text{char}}}$$

$$a_{\text{char}} = \frac{eE_{\text{char}}}{m_e}$$

Equations of Vlasov solver can be normalized as below.

$$\frac{\partial \hat{f}}{\partial \hat{t}} + v \frac{\partial \hat{f}}{\partial \hat{x}} + \hat{a} \frac{\partial \hat{f}}{\partial \hat{v}} = 0. \quad (3-5)$$

$$\frac{d^2 \hat{\varphi}(\hat{x})}{d\hat{x}^2} = -\left(1 - \int_{-\infty}^{\infty} \hat{f}(\hat{x}, \hat{v}, \hat{t}) d\hat{v}\right) \quad (3-6)$$

$$\hat{E}(\hat{x}) = -\frac{d\hat{\varphi}(\hat{x})}{d\hat{x}} \quad (3-7)$$

$$\hat{a}(\hat{x}) = -\hat{E}(\hat{x}) \quad (3-8)$$

The hat of variables means normalized variables.  $\hat{a}$  is normalized acceleration,  $\hat{E}$  is normalized electric field,  $\hat{\varphi}$  is the normalized electric potential,  $\hat{\rho}$  is the normalized number density,  $\hat{f}$  is normalized distribution function. As a result, the charge density should be calculated first, and the second is gotten electric potential. Finally, we can solve the Poisson's equation and know the electric field and acceleration of plasma itself.

### 3.3 Simulation order of Vlasov solver

The order of simulation in solving equations above in Vlasov solver is shown in figure 3-1.

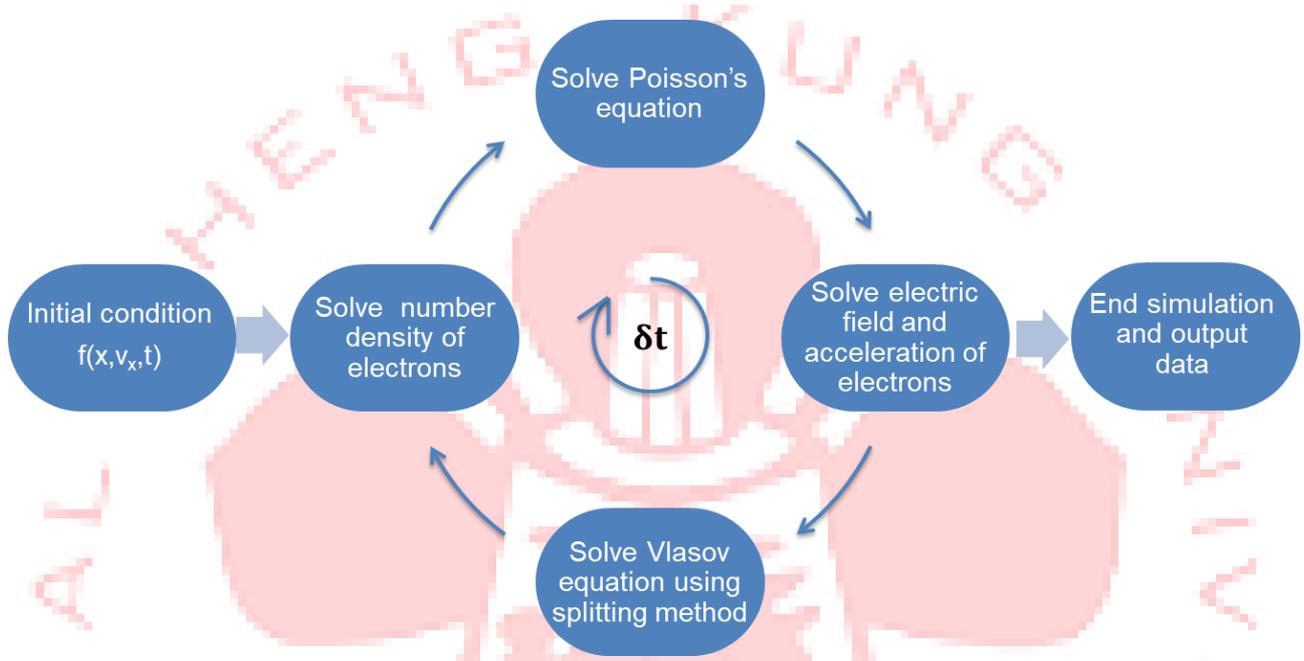


Fig.3-1 Flow chart of Vlasov solver

### 3.4 Numerical structure

The structure of Vlasov solver is shown as below.

Main program

- Main
  - Module
  - Initial
  - Boundary
  - Density
  - Poisson
  - vboundary
  - Electric
  - vboundary

- for time loop
  - Splitv
  - Boundary
  - Splitx
  - Boundary
  - Density
  - Poisson
  - vboundary
  - Electric
  - vboundary
- Output

The function of every subprogram is discussed below.

Main – main program, the time steps are set here

Module – set numerical variables

Initial – set initial distribution function

Boundary and vboundary – set boundary conditions of simulation

Density – calculate number density

Poisson – calculate Poisson's equation

Electric – calculate electric field by using electric potential relation

Splitx – calculate divided advection equation in x

Splitv - calculate divided advection equation in v

### 3.5 Numerical space grids

The grid used in the Vlasov solver is a phase space grid. It has the form shows in figure 3-2.

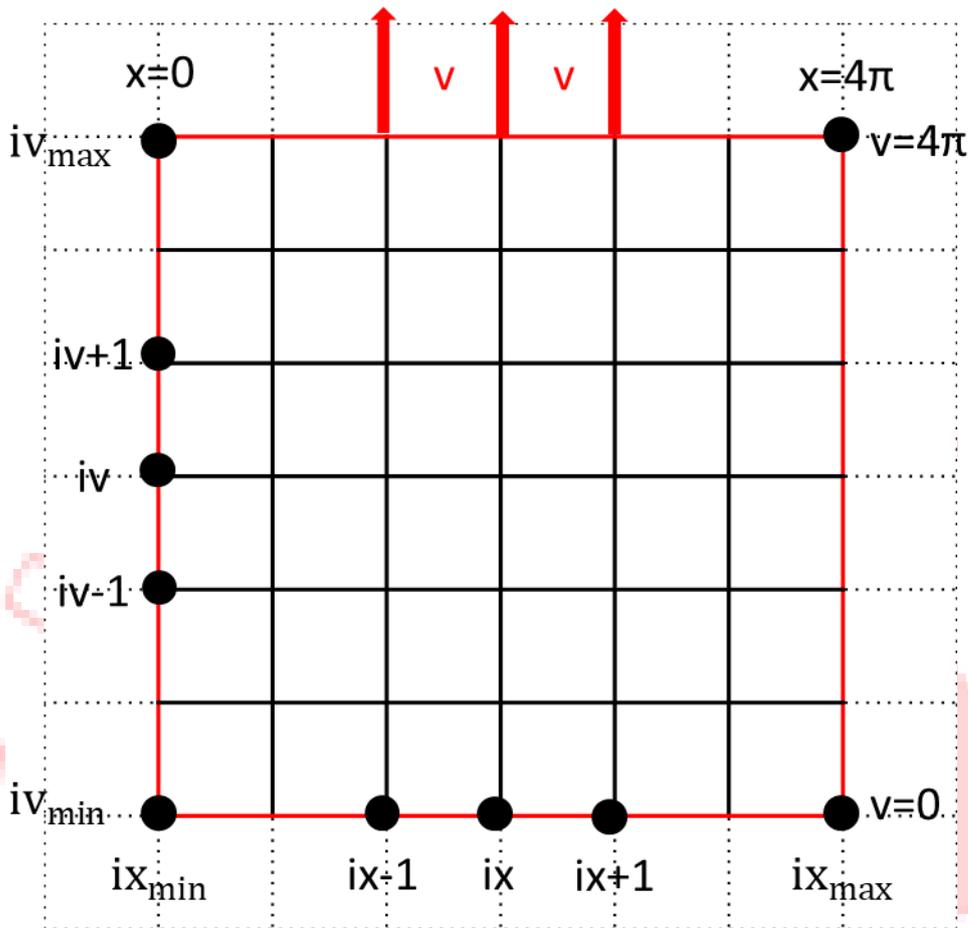


Fig.3-2 Simulation range of Vlasov solver in stencil diagram

The simulation range of Vlasov solver is set between 0 to  $4\pi$  in both  $x$  and  $v$ , and grids of  $x$  and  $v$  are both divided in 512 grids. The total simulation time is 1 with each time step is  $\frac{1}{1600}$ , and the number of total time steps should be 1600.

## Chapter 4 Numerical methods and simulation results

In this chapter, the numerical methods of each subprogram are explained and benchmarked so that we can use this Vlasov solver to solve more complex problems. To benchmark the every subprogram, only the subprogram we want to benchmark is activated.

## 4.1 Initial condition

The first initial condition is set as  $\hat{f}(\hat{t} = 0) = e^{-\frac{(\hat{x}-2\pi)^2}{2}}$ , and the graph is shown in figure 4-1. The result in red dots is compared with the analytical one in blue line which is calculated by Mathematica.

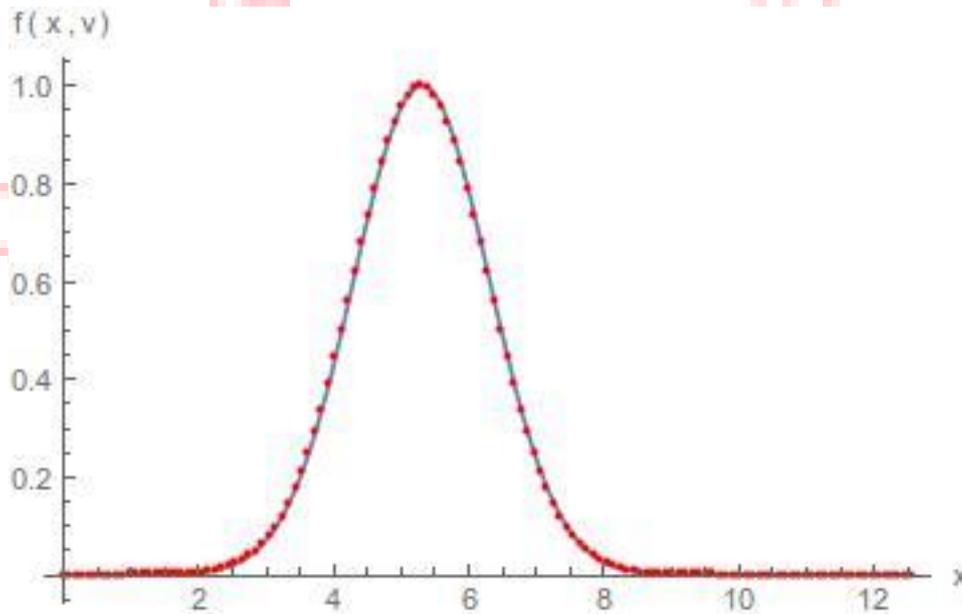
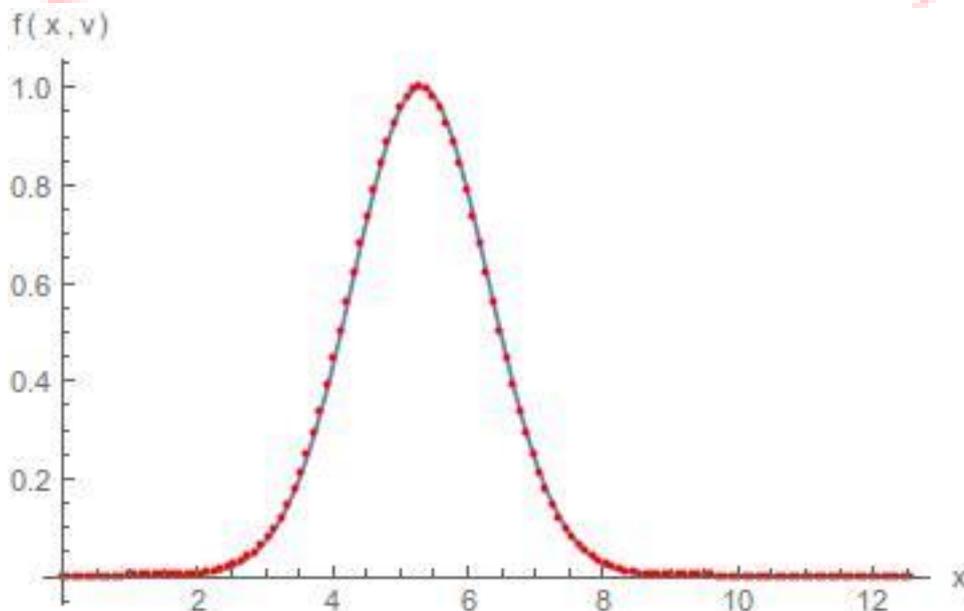


Fig.4-1 Simulation result of initial condition

The second initial condition is set as  $\hat{f}(\hat{t} = 0) = e^{-\frac{(\hat{v}-2\pi)^2}{2}}$ , and the graph is shown in figure 4-2. The result in red dots is compared with the analytical one in blue line which is calculated by Mathematica.



4-2 Simulation result of initial condition

Both figure 4-1 and figure 4-2 shows that the initial conditions of simulation are the same with the analytical one.

## 4.2 Boundary condition

The boundary condition is set as 1 in on boundary as  $v_{\min} = v_{\max} = 1$ . The initial condition is set as below and shown in figure 4-3.

$$f(x, v, t = 0) = \frac{2}{7\sqrt{2\pi}} (1 + 5v^2) \left[ 1 + 0.01 \left( \frac{\cos x + \cos 1.5x}{1.2} + \cos 0.5x \right) \right] e^{-\frac{v^2}{2}}$$

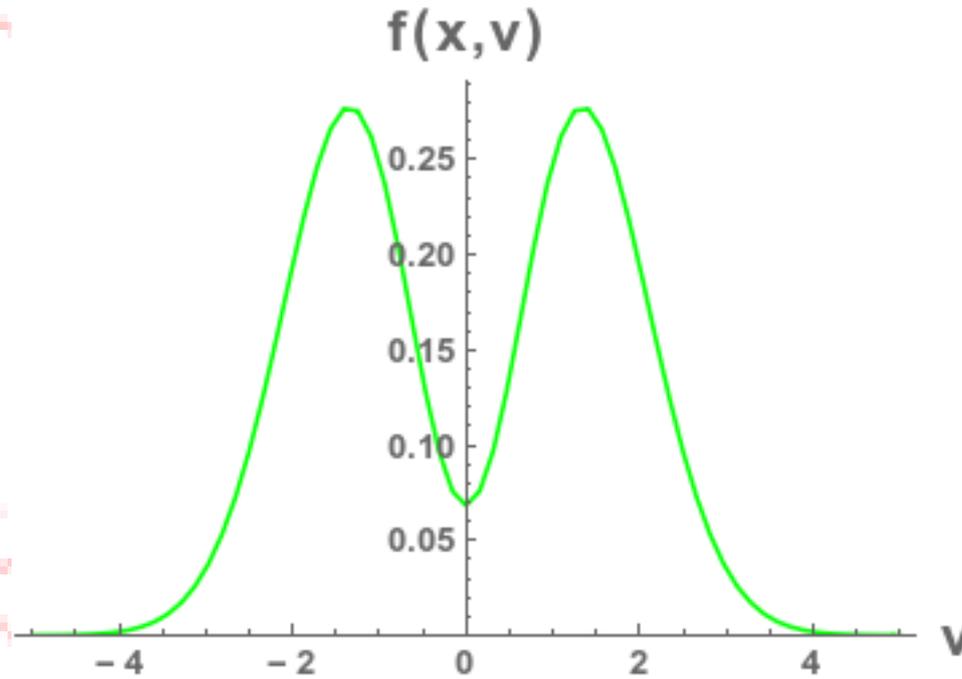


Fig.4-3 initial condition

After putting the boundary condition subprograms, the result is shown in figure

4-4.

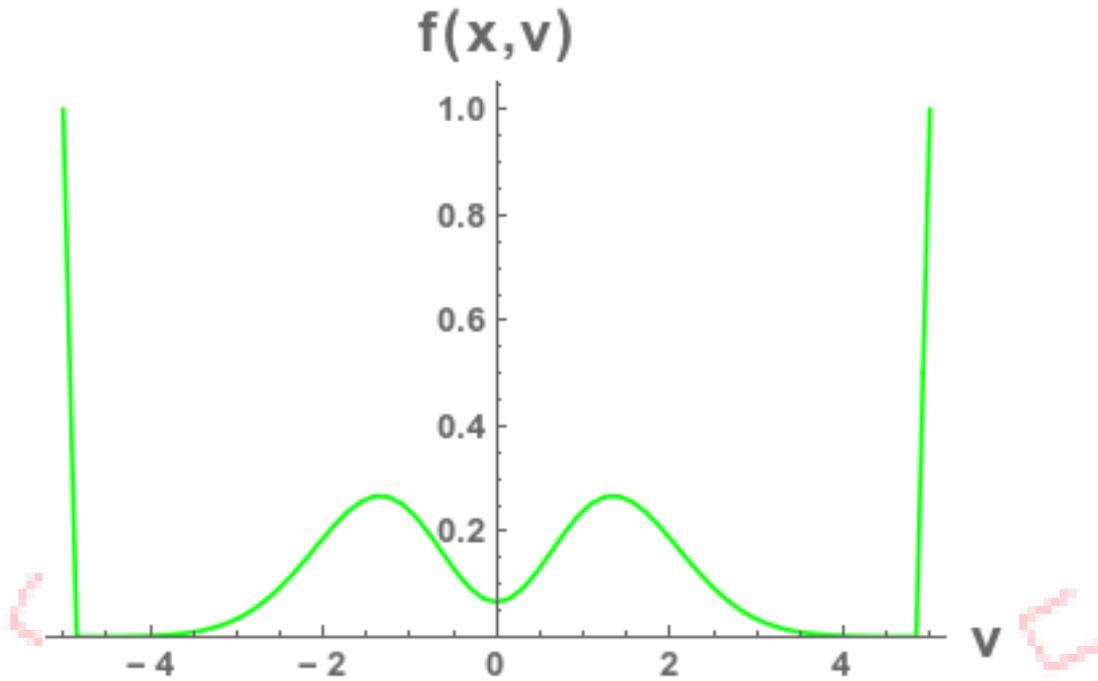


Fig.4-4 initial condition with  $v_{\min} = v_{\max} = 1$

### 4.3 Split Vlasov equations

The normalized Vlasov equation can be written as Eq.(3-5) like following.

$$\frac{\partial \hat{f}}{\partial t} + \hat{v} \frac{\partial \hat{f}}{\partial \hat{x}} + \hat{a} \frac{\partial \hat{f}}{\partial \hat{v}} = 0. \quad (3-5)$$

First, to solve this equation numerically, Vlasov equation can be split into two advection equations as the form which make it become easier in numerical simulation.

$$\frac{\partial \hat{f}}{\partial t} + \hat{v} \frac{\partial \hat{f}}{\partial \hat{x}} = 0, \quad (4-1)$$

$$\frac{\partial \hat{f}}{\partial t} + \hat{a} \frac{\partial \hat{f}}{\partial \hat{v}} = 0. \quad (4-2)$$

Then the advection equation, Eq.(4-1) and Eq.(4-2) can be discretized as

$$\frac{f_{ix}^{n+1} - f_{ix}^{n*}}{\delta t} + v \frac{f_{ix}^n - f_{ix-1}^n}{\delta x} = 0, \quad (4-3)$$

$$\frac{f_{iv}^{n*} - f_{iv}^n}{\delta t} + a \frac{f_{iv}^n - f_{iv-1}^n}{\delta v} = 0. \quad (4-4)$$

Thus make Vlasov equation become two advection equations, and then these two equations can be changed to as

$$\hat{f}_{ix}^{n+1} = \hat{f}_{ix}^{n*} - \frac{\delta t}{\delta x} \hat{v}(\hat{f}_{ix}^n - \hat{f}_{ix-1}^n), \quad (4-5)$$

$$\hat{f}_{iv}^{n*} = \hat{f}_{iv}^n - \frac{\delta t}{\delta v} \hat{a}(\hat{f}_{iv}^n - \hat{f}_{iv-1}^n). \quad (4-6)$$

where  $n^*$  is a temporal time step between  $n$  and  $n+1$  in discretization. As a result,  $\hat{f}_i^{n*}$  term can be calculated first here, and the result is used to calculate the  $\hat{f}_i^{n+1}$  at last. The split advection equations in  $x$  and in  $v$  are separated into two subprograms in this Vlasov solver. The initial condition is set as section 4-1, and the boundary conditions are periodic boundary condition on both side in  $x$ , reflective boundary condition on the bottom of  $v$  and zero on the top of  $v$ . Both of these equations can be solved like any advection equation which has been discussed in Part I, and they can be benchmarked respectively.

#### 4.3.1 Split advection equation in $x$ direction

The Split advection equation in  $x$  direction is solved by using PPM scheme in section 2.2.4, and velocity is set as  $v = 1$ . The initial distribution function is  $e^{-\frac{(x-2\pi)^2}{2}}$ . The simulation result of it is shown in figure 4-5 as solid line which is compared with Mathematica as red dots. It shows the wave moves from left to right well.

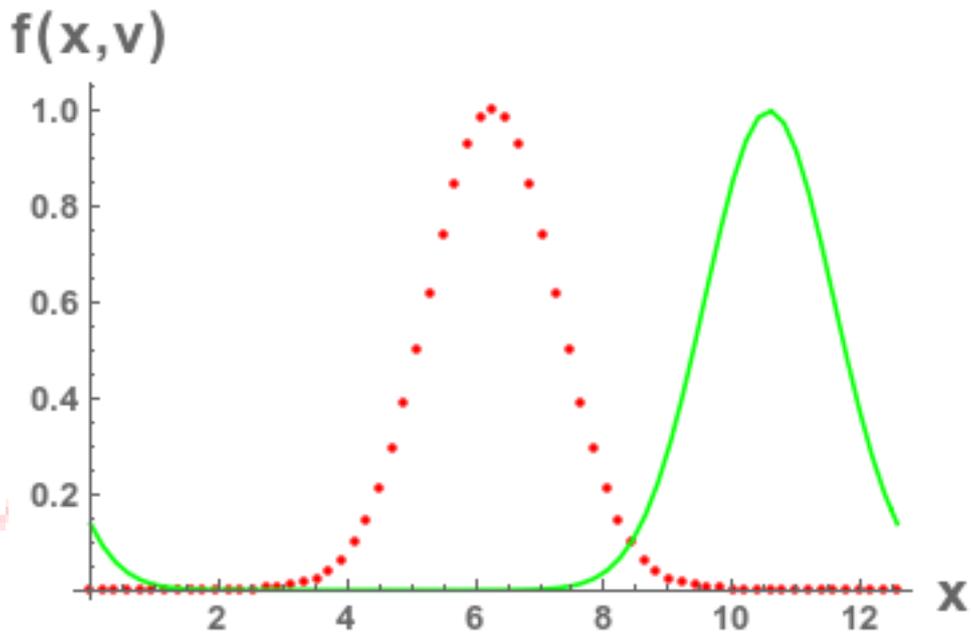


Fig.4-5 simulation result of split advection equation in x direction

#### 4.3.2 Split advection equation in v direction

The split advection equation in x direction is also solved by using PPM scheme in section 3.2.4, and acceleration is set as  $a = 1$ . The simulation result of it is shown in figure 4-6 as solid line which is compared with Mathematica as red dots. It shows the wave moves from left to right well.

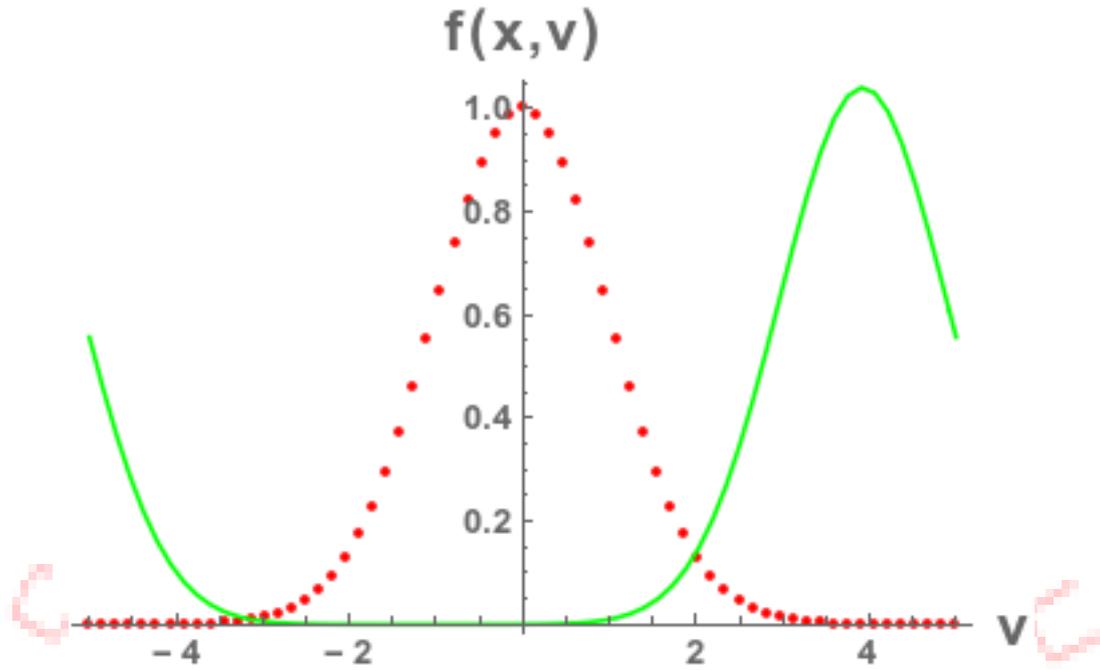


Fig.4-6 simulation result of split advection equation in v direction

#### 4.4 Charge density

The charge density is calculated from the integration of distribution function  $\hat{f}$  as Eq.(4-7).

$$\hat{\rho} = 1 - \int \hat{f} d\hat{v}. \quad (4-7)$$

To benchmark it, we set the initial distribution function  $\hat{f}$  as

$$\hat{f}(\hat{t} = 0) = \frac{1}{1 + \hat{v}^2}.$$

The trapezoidal method is used to calculate this integration, so  $\hat{\rho}$  can be calculated by using

$$\hat{\rho} = \int_{v_{imin}}^{v_{imax}} \hat{f} d\hat{v} = \frac{v_{imax} - v_{imin}}{iv_{max}} \left[ \sum_{iv=1}^{iv_{max}-1} f(v_{iv}) + \frac{1}{2} (f(v_{ivmin}) + f(v_{ivmax})) \right]. \quad (4-8)$$

The integration of distribution function by velocity is shown in figure 4-7 to compare with analytical one. The red points are simulation result, and solid line is analytical one.

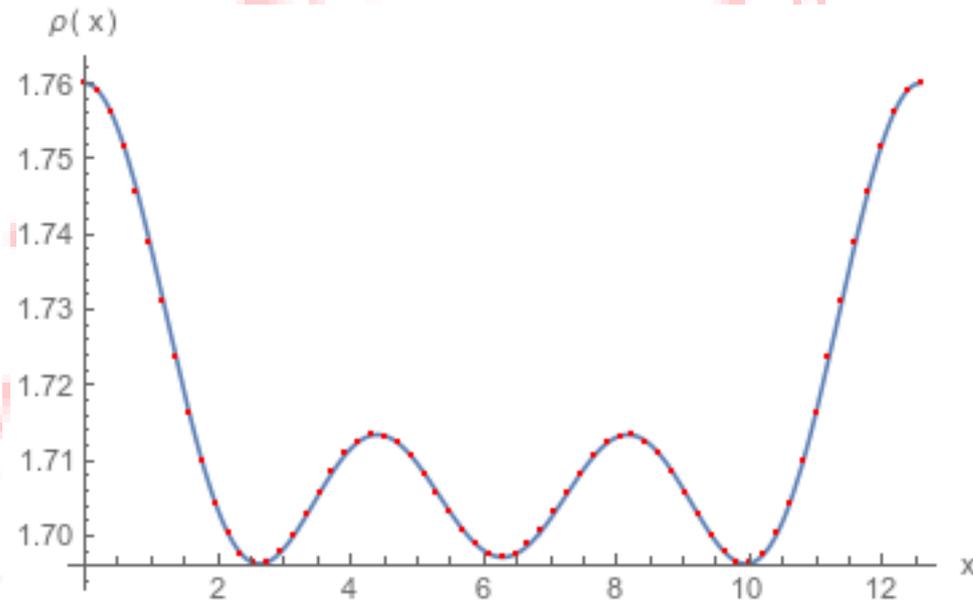


Fig.4-7 simulation result of number density

#### 4.5 Poisson's equation

The Poisson's equation has the form as Eq.(3-6).

$$-\frac{d^2\hat{\varphi}}{d\hat{x}^2} = 1 - \hat{\rho} = 1 - \int \hat{f}d\hat{v}. \quad (3-6)$$

To benchmark this Poisson's equation, we set  $\hat{\rho} = \sin\left(\frac{3\hat{x}}{4\pi}\right)$ , so the equation is set as

$$\frac{d^2\hat{\varphi}}{d\hat{x}^2} = \sin\left(\frac{3\hat{x}}{4\pi}\right). \quad (4-9)$$

For testing of this subprogram,  $\rho(x) = \sin\left(\frac{3x}{4\pi}\right)$ . The boundaries are set as  $\varphi(0) = 0, \varphi(4\pi) = 0$ .

The exact solution of Eq.(4-9) is

$$\hat{\phi} = \frac{4}{9} [\hat{x}\pi\sin(3) - 4\pi^2\sin(\frac{3\hat{x}}{4\pi})].$$

This solution can be used in comparing with the numerical one solved in following.

To solve Poisson equation numerically, Eq.(4-9) can be discretized to the finite difference equation from.

$$\frac{-\phi_{ix-1} + 2\phi_{ix} - \phi_{ix+1}}{\delta x^2} = \rho_{ix} \quad (4-10)$$

This equation can be changed the form to following form.

$$-\phi_{ix-1} + 2\phi_{ix} - \phi_{ix+1} = \rho_{ix} \delta x^2. \quad (4-11)$$

Eq.(4-11) can be expanded in a matrix form in different numerical grids as

$$\begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 2 & -1 & \ddots & \ddots & \vdots \\ -1 & 2 & \ddots & 2 & -1 \\ \vdots & \ddots & \ddots & -1 & 2 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi_0 \\ \vdots \\ \vdots \\ \vdots \\ \phi_{imax} \end{bmatrix} = \begin{bmatrix} \rho_0 \delta x^2 \\ \vdots \\ \vdots \\ \vdots \\ \rho_{imax} \delta x^2 \end{bmatrix}. \quad (4.12)$$

There are three methods, (a) Jacobi's method, (b) Gauss-Seidel's method, (c) Tridiagonal method, are used to solve this matrix, where (a) and (b) are iteration methods, and (c) is direct methods for solving matrix.

(a) Jacobi's method

The Poisson's equations in Eq.(4-12) are solved iteratively by using the following form as Eq.(4-13).

$$\phi_{ix}^{n+1} = 0.5(\phi_{ix-1}^n + \rho_{ix}^n \delta x^2 + \phi_{ix+1}^n). \quad (4-13)$$

(2) Gauss-Seidel's method

Like the iteration method in Jacobi's method, the Poisson's equations in Eq.(4-12) are solved iteratively by using the following form as Eq.(4-14).

$$\varphi_{ix}^{n+1} = 0.5(\varphi_{ix-1}^{n+1} + \rho_{ix}^n \delta x^2 + \varphi_{ix+1}^n). \quad (4-14)$$

Comparing to Jacobi's theorem, the  $\varphi_{ix-1}^{n+1}$  terms are used n+1 temporal step for ix-1 space grid, and this way makes its convergence fast.

### (3) Tridiagonal method

The tridiagonal method is used to solve the tridiagonal matrix, which use the Gaussian elimination to solve the matrix. The steps of tridiagonal method in solving 1-D Poisson's equation are shown here.

$$\begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 2 & -1 & \ddots & \ddots & \vdots \\ -1 & 2 & \ddots & 2 & -1 \\ \vdots & \ddots & \ddots & -1 & 2 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} \varphi_0 \\ \vdots \\ \vdots \\ \vdots \\ \varphi_{ixmax} \end{bmatrix} = \begin{bmatrix} b_0 & c_0 & 0 & \dots & 0 \\ a_1 & b_1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & b_{ixmax-1} & c_{ixmax-1} \\ 0 & \dots & 0 & a_{ixmax} & b_{ixmax} \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ \vdots \\ \vdots \\ x_{ixmax} \end{bmatrix} = \begin{bmatrix} \rho_0 \delta x^2 \\ \vdots \\ \vdots \\ \vdots \\ \rho_{ixmax} \delta x^2 \end{bmatrix} = \begin{bmatrix} d_0 \\ \vdots \\ \vdots \\ \vdots \\ d_{ixmax} \end{bmatrix}. \quad (4-15)$$

By using Gaussian elimination, Eq.(4-15) can be changed to

$$\begin{bmatrix} b'_0 & c'_0 & 0 & \dots & 0 \\ 0 & b'_1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & b'_{ixmax-1} & c'_{ixmax-1} \\ 0 & \dots & 0 & 0 & b'_{ixmax} \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ \vdots \\ \vdots \\ x_{ixmax} \end{bmatrix} = \begin{bmatrix} d_0 \\ \vdots \\ \vdots \\ \vdots \\ d_{ixmax} \end{bmatrix}. \quad (4-16)$$

$$\begin{bmatrix} b''_0 & 0 & 0 & \cdots & 0 \\ 0 & b''_1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & b''_{ixmax-1} & 0 \\ 0 & \cdots & 0 & 0 & b''_{ixmax} \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ \vdots \\ \vdots \\ x_{ixmax} \end{bmatrix} = \begin{bmatrix} d'_0 \\ \vdots \\ \vdots \\ \vdots \\ d'_{ixmax} \end{bmatrix}. \quad (4-17)$$

This matrix can be changed as the iteration form.

$$\sum_{i=1}^{ixmax} c'_i = \frac{c_i}{b_i - a_i c'_{i-1}}. \quad (4-18)$$

$$\sum_{i=1}^{ixmax-1} d'_i = \frac{d_i - a_i d'_{i-1}}{b_i - a_i c'_{i-1}}. \quad (4-19)$$

$$\sum_{i=ixmax-1}^0 x_i = d'_i - c'_i x_{i+1}. \quad (4-20)$$

All of the three methods are shown in almost the same results. The figure 4-8 shows the simulation result is compared with the analytical one.

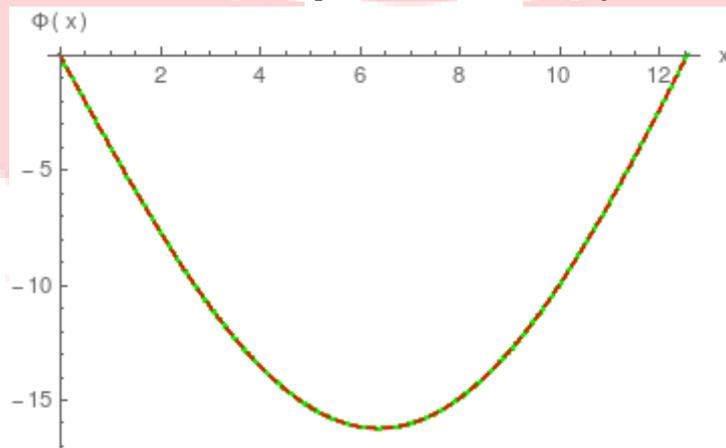


Fig.4-8 simulation result of Poisson's equation method

#### 4.4 Electric field and acceleration

The relationship between electric potential and electric field is shown by Eq.(3.7).

$$\hat{E}(\hat{x}) = -\frac{d\hat{\phi}(\hat{x})}{d\hat{x}} \quad (3-7)$$

Electric potential  $\hat{\phi}$  is obtained from the result of Poisson's equation. To benchmark this subprogram, the result above in section 4.4 and 4.5 of Poisson's

equation, and charged density can be used, the initial distribution function is also the same as above. Analytical solution of electric field is

$$\hat{E} = -\frac{4}{9}(-\pi\hat{x}\text{Sin}[3] + 4\pi^2\text{Sin}[\frac{3\hat{x}}{4\pi}])$$

To solve Eq.(3-7) numerically, differential equation can be discretized by using central difference method as the from

$$E_{ix} = -\frac{\phi_{ix+1} - \phi_{ix-1}}{2\delta x} \tag{4-21}$$

The simulation result of Eq.(4-21) is shown in figure 4-9.

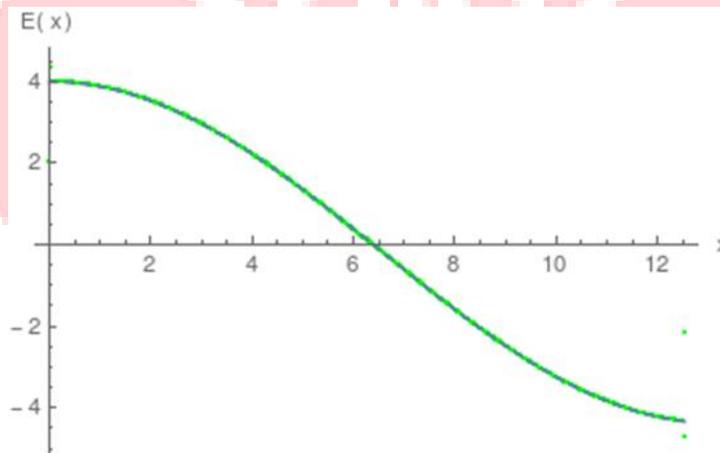


Fig.4-9 simulation result of Electric field

## Chapter 5 Future work

### 5.1 Short term goal

In recent, the 1-D Vlasov solver will be used to solve the two-stream instability.

The initial distribution function  $f(x, v, t = 0) = \frac{2}{7\sqrt{2\pi}}(1 + 5v^2) [1 + 0.01(\frac{\cos x + \cos 1.5x}{1.2} + \cos 0.5x)]e^{-\frac{v^2}{2}}$  [2] will be used. The result will be compared with the one simulated by Particle-in-Cell (PIC) method, a basic model in kinetic regime plasma phenomena for simulating.

### 5.2 Long term goal

After the 1-D Vlasov solver is benchmarked, the 2-D Vlasov solver with distribution function  $f(x, y, v_x, v_y, t)$  will be further implemented. The 2-D Vlasov-Poisson system will be benchmarked with fluid transportation in a 2-D phase space.

## Chapter 6 Summary

After practicing some numerical schemes to solve advection equation in numerical hydrodynamics, the basic practicing of numerical simulation has been enough. The most precise method among these methods in solving advection equations is Piecewise parabolic method. This method is also used in the Vlasov solver in Part II. The subprograms of Vlasov solver including initial condition, boundary condition, splitting equation of  $x$  and  $v$ , density, electric field, acceleration, and Poisson's equation have been benchmarked in 1-D Vlasov-Poisson system, the next step is going to simulate two-stream instability phenomena in plasma.

## Reference

[1] Bernd Freytag, Introduction to Numerical Hydrodynamics, January 24, 2010

([http://www.astro.uu.se/~bf/course/numhd\\_course\\_20100124.pdf](http://www.astro.uu.se/~bf/course/numhd_course_20100124.pdf))

[2] F. Filbet, E. Sonnendrücker, Comparison of Eulerian Vlasov solvers, Comp. Phys.

Comm. 150, 247–266(2003)

