National Cheng Kung University
Institute of Space and Plasma Physics

Biannual Progress Report

研究生：楊宗桓

指導教授：張博宇　博士

中華民國一○六年二月二十八日星期二

摘要

此報告將會介紹利用數值模擬的方法來解基本流體力學方程式。在我們模擬更複雜的太空電漿現象之前，我們先練習利用數值模擬的方法解一些簡單的偏微分方程。我們參考“Introduction to Numerical Hydrodynamics”這本書上的例題，並且練習書上的流體方程式，並將我們模擬的結果和書上的結果作比較來檢查我們對這些方法是否了解。

Abstract

In this report, different numerical methods for solving the basic equations in hydrodynamics are introduced. We would like to practice solving simple ordinary differential equations (ODEs) and partial differential equations (PDEs) before simulating complicated phenomena in space. We are following examples in the textbook "Introduction to Numerical Hydrodynamics." Simulation results are compared to the results on the textbook to verify that we know how to use different scheme on solving PDEs.

Content

Chapter 1 Introduction

1.1 Motivation

We would like to use numerical schemes to study the behaviors of plasma in the solar-terrestrial system, which is in the kinetic regime. There are some complex equations that need be understood. The equations can be solved numerically and the practical problems can be studied. Before solving the equations for plasma, we follow the textbook, "Introduction to Numerical Hydrodynamics" [1] to get familiar with varies numerical schemes to solve different equations. The details will be given in the following sections.

1.2 Background

The space plasma is in the kinetic regime and can be described by using the following equations.

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{t}} + \boldsymbol{v}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \boldsymbol{a}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}} = \boldsymbol{0}, \text{ Vlasov equation}$$

$$\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{t}} + \nabla \cdot \mathrm{J} = \boldsymbol{0}, \text{ continuity equations}$$

$$\begin{cases} \nabla \times \mathrm{B} = \mu_0 J + \frac{1}{c^2}\frac{\partial E}{\partial t} \\ \nabla \times \mathrm{E} = -\frac{\partial B}{\partial t} \\ \nabla \cdot \mathrm{B} = 0 \\ \nabla \cdot \mathrm{E} = \frac{\rho}{\epsilon_0} \end{cases} \text{, Maxwell equations}$$

To solve these equations numerically, the equations need to be discretized. For example, the Vlasov equation above in the difference form can be written in the discrete form.

$$\frac{df_{i,j}^{n+1}-df_{i,j}^{n}}{\delta t}+\mathbf{v}\frac{df_{i+1,j}^{n}-df_{i-1,j}^{n}}{2\delta x}+\mathbf{a}\frac{df_{i,j+1}^{n}-df_{i,j-1}^{n}}{\delta v}=\mathbf{0},$$

We would like to practice in solving simple differential equations numerically before solving the equations above.

1.3 The textbook “Introduction to Numerical Hydrodynamics”

This textbook includes many different numerical methods in computational fluid dynamics. It not only gives different numerical schemes but also provides lots of corresponding results. It is a good textbook for us to get familiar with different numerical schemes. It is divided into two parts: (1) solving linear PDE; (2) nonlinear PDE. They give some discussions and introductions of numerical basic concepts and hydrodynamics equations that beyond the range of this report. In this report, the linear PDEs are solved numerically and the results are given.

Chapter 2 Simulation setups

The most conventional equations in fluid and kinetic theory are differential equations, which include two types, ordinary differential equations (ODEs) and partial differential equations (PDEs). The partial differential equations can also be categorized in three kinds, hyperbolic, elliptical, and parabolic PDEs. In this chapter, there are some equations will be given as examples. They will be discretized via finite difference method. The initial and boundary conditions will also be given. The results of solving these equations numerically will be given in the next chapter.

2.1 Basic concepts of numerical simulation

The basic concepts of solving differential equations numerically include discretizing differential equation, grid generation, iteration, and numerical errors, etc. Some of them will be discussed here, and the others will be discussed in the next chapter.

2.1.1 Discretization

The differential equations need to be discretized by finite difference method (FDM) to be solved numerically, both in space and in time. Also, the method can be used in time or other physical quantities. There are three basic finite difference methods as the form.

$\frac{d}{dx}f(x) \to \frac{f(x+\delta x)-f(x-\delta x)}{2\delta x}$ Central difference,

$\frac{d}{dx}f(x) \to \frac{f(x+\delta x)-f(x)}{\delta x}$ Forward difference,

$\frac{d}{dx}f(x) \to \frac{f(x)-f(x-\delta x)}{\delta x}$ Backward difference,

2.1.2 Grid generation

The generation of grids is to transfer real time-space to numerical time-space. There are several kinds of ways to define grids in space, time, and other parameters following.

1. Eulerian method: grids are fixed in space.
2. Lagrangian method: grids are moving with the flow.
3. Semi-Lagrangian method: this method is based on Eulerian one, but the grids move with speed different from the flow.
4. Grids are replaced by particle positions, which didn't use any grids.

2.1.3 Stencil diagram

Stencil diagram is a kind of graph to represent the relations between grids in the new time step to grids in the old time step.

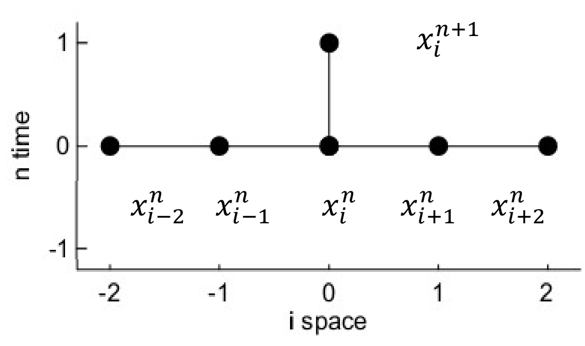

Fig.1 Stencil diagram

In figure 1, the horizontal axis represents the space grids, and vertical axis represents the time grids. This graph represents that one grid in new time step depends on the same point and four neighbor points in the old time steps (five points in total).

2.2 Basic equations

Discretization of ODE and PDE will be given in this section. We will introduction simple ODE and three kinds of PDE, diffusion, linear advection, and Poisson equations in the following. The initial and boundary condition can be used in these equations will also be discussed.

2.2.1 Ordinary differential equation (ODE)

The simple linear ordinary differential equation is the differential equation which has only one independent variable, it can be shown as the following form.

$$\frac{dy}{dt} = -a * y, \text{ where “a” is a constant}$$

By using finite difference method, the equation can be discretized to the form.

$$\frac{y^{n+1} - y^{n}}{dt} = -a * y^{n}.$$

And it can be derived to

$$y^{n+1} = y^{n} - a * y^{n} * dt.$$

Here, time is also discretized to

$$t^n = dt * n + t_0.$$

2.2.2 Parabolic PDE

The parabolic equation is also called diffusion equation or heat transfer equation, it describes heat transfer and diffusion phenomena. There are two independent variables in this equation and has the form.

$$\frac{\partial f}{\partial t} = \kappa \frac{\partial^2 f}{\partial x^2}.$$

The equation can be discretized by using FDM.

$$\frac{\partial f}{\partial t} \to \frac{f_i^{n+1} - f_i^n}{\delta t},$$

$$\kappa \frac{\partial^2 f}{\partial x^2} \to \kappa \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\delta x^2}.$$

Therefore, the equation can be discretized as below.

$$\frac{f_i^{n+1} - f_i^n}{\delta t} = \kappa \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\delta x^2}.$$

Finally, the parabolic PDE can be written in the form as below.

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

The time and space are discretized as below.

$$t^n = dt * n + t_0,$$

$$x_i = dx * i + x_0.$$

2.2.3 Hyperbolic PDE

The hyperbolic equation is also called advection equation. This equation is used to express the advection flow in physics and has the form.

$$\frac{\partial f}{\partial t} = v\frac{\partial f}{\partial x}.$$

By applying finite difference method, it becomes

$$\frac{\partial f}{\partial t} \to \frac{f_i^{n+1} - f_i^n}{\delta t},$$

$$v\frac{\partial f}{\partial x} \to v\frac{f_{i+1}^n - f_{i-1}^n}{2\delta x}.$$

The equation becomes

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

As a result, the hyperbolic PDE is in the form.

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

The time and space are also discretized as below,

$$t^n = dt * n + t_0,$$

$$x_i = dx * i + x_0.$$

2.2.4 Elliptical PDE

Finally, the last kind of partial differential equation is elliptical equation, such as Laplace's equation and Poisson's equation.

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0 \text{ for Laplace's equation,}$$

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \rho \text{ for Poisson's equation.}$$

These two equations are usually used to represent potential fields in physics. By applying finite difference method, the terms of these two equations become

$$\frac{\partial^2 f}{\partial x^2} \rightarrow \frac{f_{i+1}^{j}-2f_i^j+f_{i-1}^{j}}{\delta x^2},$$

$$\frac{\partial^2 f}{\partial y^2} \rightarrow \frac{f_i^{j+1}-2f_i^j+f_i^{j-1}}{\delta y^2}.$$

So the equations becomes

$$\frac{f_{i+1}^{j}-2f_i^j+f_{i-1}^{j}}{\delta x^2} + \frac{f_i^{j+1}-2f_i^j+f_i^{j-1}}{\delta y^2} = \rho_i^j.$$

As a result, the elliptical PDE is in the form.

$$\frac{f_i^{j+1}+f_{i+1}^{j}-4f_i^j+f_{i-1}^{j}+f_i^{j-1}}{h^2} = \rho_i^j,$$

$$h^2 = \delta x^2 = \delta y^2.$$

Finally, the equation has the form.

$$f_i^{j^*} = \frac{1}{4}(f_i^{j+1} + f_{i+1}^{j} - \rho_i^j h^2 + f_{i-1}^{j} + f_i^{j-1}).$$

2.2.5 Initial conditions (ICs) and boundary conditions (BCs)

To solve partial differential equations, initial conditions and boundary conditions need to be given. The initial condition defines the initial state of the function and always has the form, $y(t=0) = f(x)$, where $f(x)$ is a function of space.

The boundary condition defines the physical meaning on boundaries. There are some boundary conditions in the following.

1. Dirichlet boundary condition, $\mathrm{y}(0)$ *and* $\mathrm{y}(i_{max})$ are given, where y is the function of space, and i_{max} represents the points on the boundary.

2. Neumann condition,$y'(0)$ *and* $y'(i_{max})$, where y is the function of space, and i_{max} represents the points on the boundary..

3. Reflective boundary condition, $y(i_{max} - 1) = y(i_{max} + 1)$, $y(-1) = y(1)$, where y is the function of space, and i_{max} represents the points on the boundary.

4. Periodic boundary condition, $y(i_{max} + 1) = y(1)$, $y(-1) = y(i_{max} - 1)$, where y is the function of space, and i_{max} represents the points on the boundary.

Chapter 3 Numerical results

The ODE and PDE we have already discussed in last chapter will be solved numerically in this chapter by FDM. The positivity and stability of these numerical results will also be shown.

3.1 Positivity and stability

In solving problems numerically, the numerical noises can affect the results. In order to solve these problems, measuring the positivity and stability of numerical results can become an effective way for us to estimate its accuracy. Here, we will verify the positivity and stability of ODE and PDE respectively.

3.1.1Simple ordinary differential equation

The equation $\frac{dy}{dt} = -a * y$ as an example can be discretized as

$$y^{n+1} = y^n - a * y^n * dt.$$

If "a" in the equation is 1, the criterion of the numerical noise is

Criterion for stability: $dt < \frac{2}{a}$,

Criterion for positivity: $dt < \frac{1}{a}$.

The results of solving the ODE numerically with different δt are given in figure 2 and figure 3.

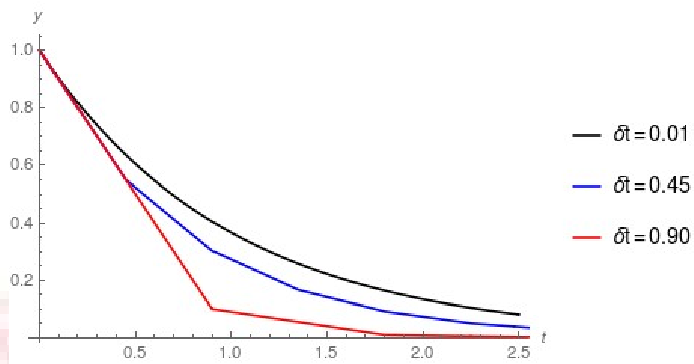

Fig.2 ODE simulation results with different time difference

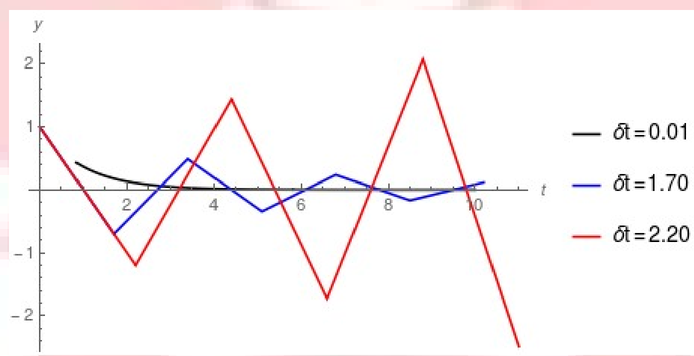

Fig.3 ODE simulation results with different time difference

Figure 2 and figure 3 show the simulation results with different dt. If dt isn't over 1, the simulation results will be always positive. Similarly, the simulation results are divergence if the time difference "dt" isn't over 2.

3.1.2 Parabolic partial differential equation

The parabolic PDE has the form as below,

$$\frac{\partial f}{\partial t} = \kappa \frac{\partial^2 f}{\partial x^2}.$$

The equation can be discretized to the form as below.

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

The criterion of stability and positivity are

Criterion for stability: $\frac{\delta t}{\delta x^2} < \frac{1}{2\kappa}$,

Criterion for positivity: $\frac{\delta t}{\delta x^2} < \frac{1}{4\kappa}$.

The initial condition is set as

$$f(x)=\begin{cases} 1, & x = 0.2,0.4,0.6,0.8,1.0 \\ 0, & x = 0.1,0.3,0.5,0.7,0.9 \end{cases}.$$

On the other hand, $\frac{\kappa\delta t}{\delta x^2} = 0.2, 0.4, 0.6$ in the figure 4, figure 5, and figure 6 respectively, while k is 1.

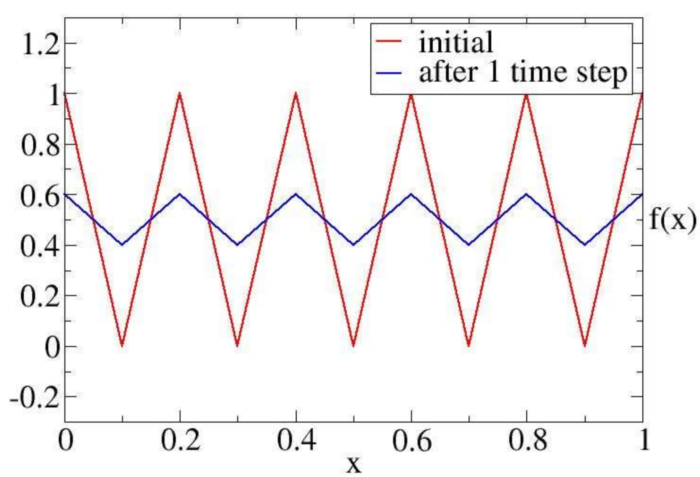

Fig.4 PDE simulation results with $\frac{\kappa\delta t}{\delta x^2}$=0.2

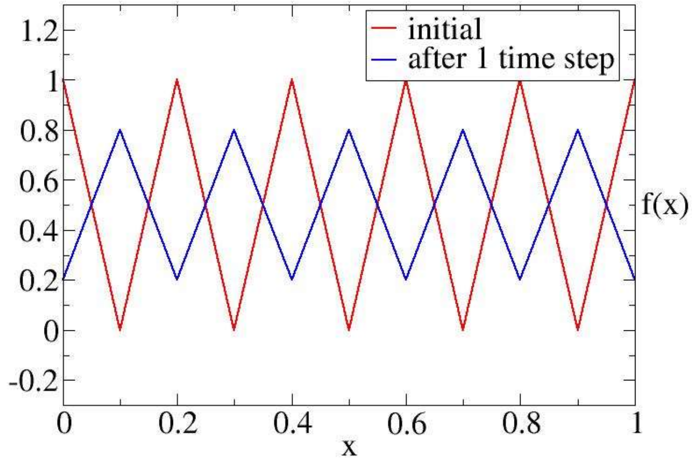

Fig.5 PDE simulation results with $\frac{\kappa\delta t}{\delta x^2}$=0.4

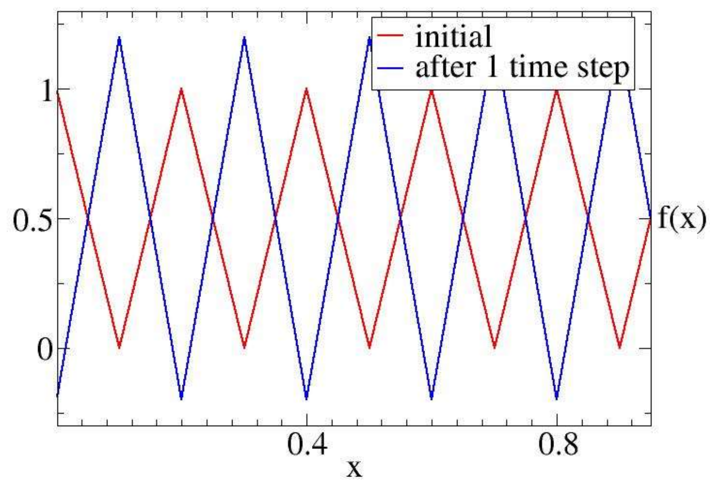

Fig.6 PDE simulation results with $\frac{\kappa\delta t}{\delta x^2}$=0.6

Figure 4, figure 5, and figure 6 show that the numerical noises depend on $\frac{\kappa\delta t}{\delta x^2}$. If $\frac{\kappa\delta t}{\delta x^2}$ is more than 0.5, the numerical results don't converge as shown in figure 6. If $\frac{\kappa\delta t}{\delta x^2}$ is less than 0.25, the numerical results are positive which is shown in figure 4.

3.2 Numerical methods verification

The PDEs which we use here are either parabolic PDE (heat equation) or hyperbolic PDE (linear advection equation). The initial condition is set as below.

0.1<x<0.2, $f(x) = e^{-\frac{(x-0.15)^2}{2\sigma^2}}$, σ=0.01 for Gaussian distribution function;

0.3<x<0.4, f(x) = 1 for rectangular function;

0.5<x<0.55, f(x) = 20x-10 and 0.55<x<0.6, f(x) = 12-20x for triangular function;

0.7<x<0.8, $f(x) = b\sqrt{1-\frac{(x-0.75)^2}{a^2}}$, b=1, a=0.75 for semi-ellipse function.

The initial condition is shown as below.

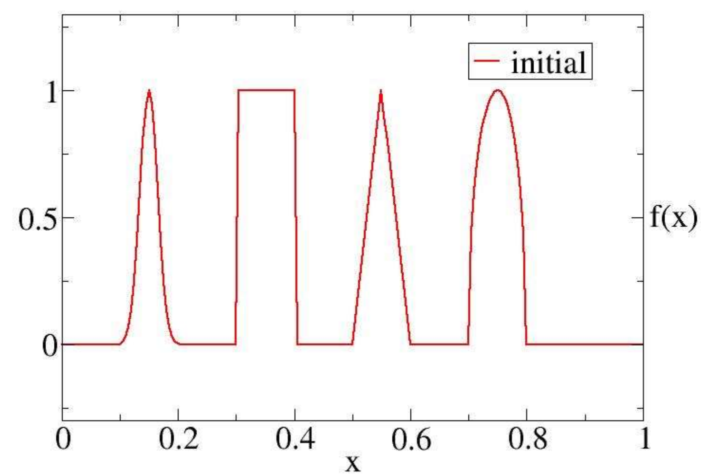

Fig.7 Initial condition

The initial condition in figure 7 includes a Gaussian function, a rectangular function, a triangular function, and a half-ellipse function from left to right, respectively. The boundary condition (B.C.) is f(-1) = f(imax-1) and f(imax+1) = f(1) as a periodic boundary condition that we already defined in section 2.2.5. The functions will propagate from right to left boundaries.

3.2.1 Parabolic partial differential equation in explicit Euler scheme

The difference form of parabolic PDE by using central finite difference is

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

The simulation results of solving the parabolic PDE which uses explicit Euler scheme are given as below. Figure 8 shows the diffusion result after 500 time steps with $\frac{\kappa\delta t}{\delta x^2} = 0.1$ and 200 space grids.

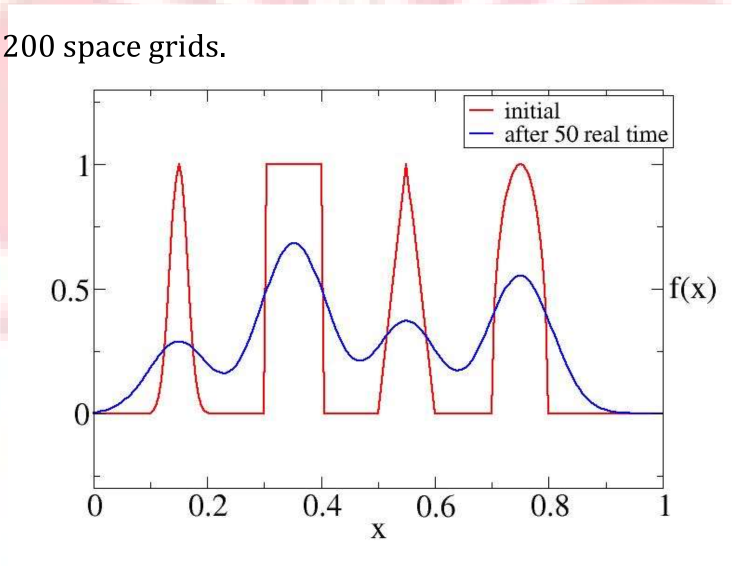

Fig.8 Diffuse equation with $\frac{\kappa\delta t}{\delta x^2} = 0.1$ and 500 time steps

Figure 9 shows the diffusion result after 100 time steps with $\frac{\kappa\delta t}{\delta x^2} = 0.5$ and 200 space grids.

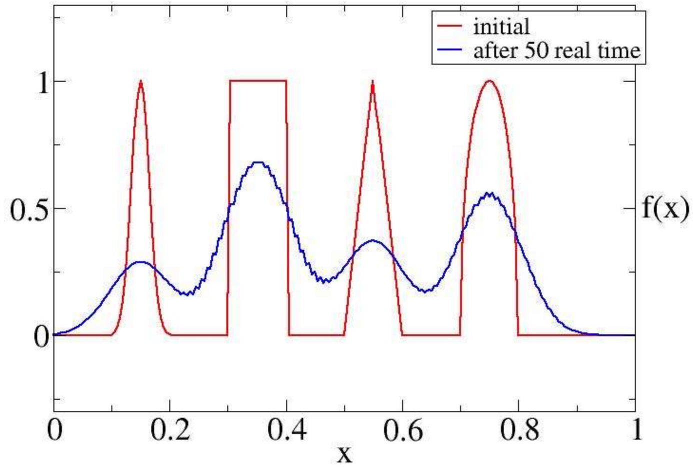

Fig.9 Diffuse equation with $\frac{\kappa\delta t}{\delta x^2} = 0.5$ and 100 time steps

The simulation results of figure 8 and figure 9 tell us that too large a $\frac{\kappa\delta t}{\delta x^2}$ causes the ripples, but the result in figure 9 is still converged.

The simulation results after 1000 time steps are shown in figure 10, the diffusion phenomena propagates very smoothly which showed that the method can be used to solve diffusion equation.

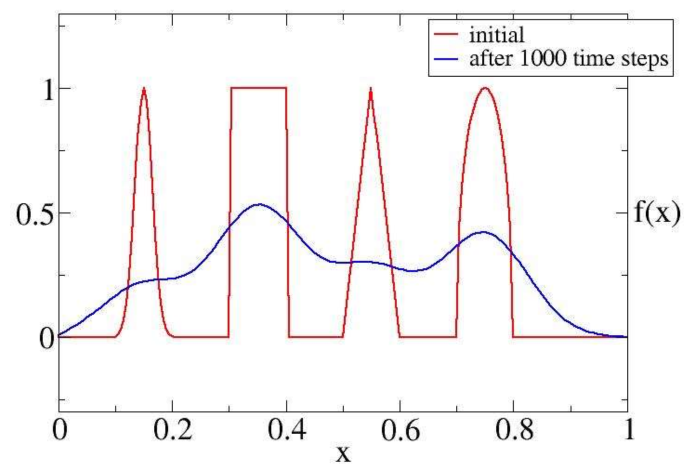

Fig.10 Diffuse equation after 1000 time steps

3.2.2 Hyperbolic partial differential equation in explicit Euler scheme

The difference form of hyperbolic PDE by using central finite difference is

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

The simulation results of the hyperbolic PDE in explicit Euler scheme are given as below. Figure 11 shows the advection result which propagates after 50 time steps with $\frac{v\delta t}{2\delta} = 0.1$ and 200 space grids.

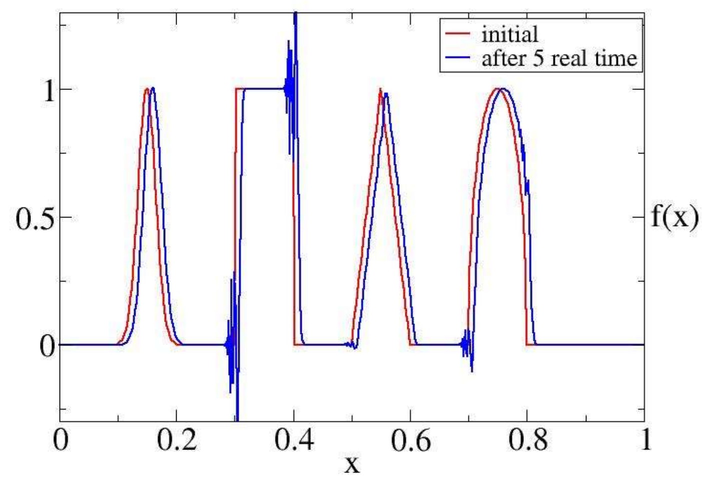

Fig.11 Advection equation with $\frac{v\delta t}{2\delta x} = 0.1$ and 50 time steps

Figure 12 shows the advection result which propagates after 10 time steps with $\frac{v\delta t}{2\delta x} = 0.5$ and 200 space grids.

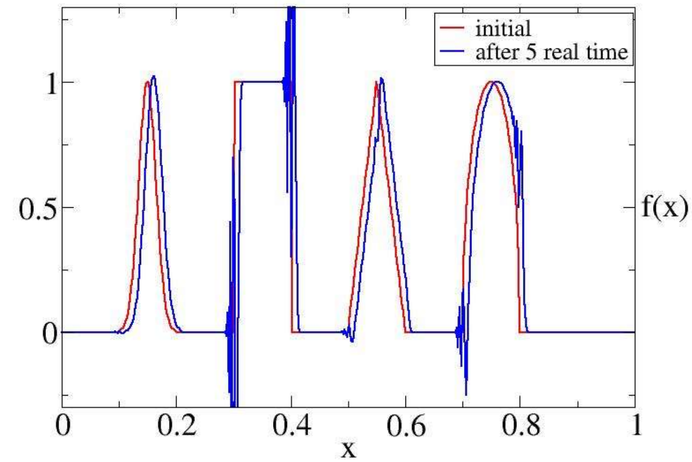

Fig.12 Advection equation with $\frac{v\delta t}{2\delta x} = 0.5$ and 10 time steps

The simulation results in figure 11 and figure 12 tell us that the growing of the oscillations makes this scheme useless.

Figure 13 shows the linear advection propagates after 1000 time steps with $\frac{v\delta t}{2\delta} = 0.1$. Serious numerical oscillations occur. Therefore, the further goal is using other numerical methods to simulate the advection equation.

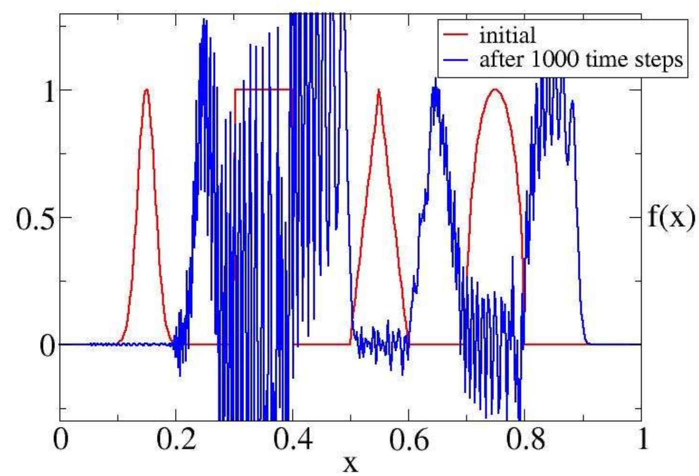

Fig.13 Advection equation after 1000 time steps

3.3 Discussions of different schemes for solving hyperbolic PDE

In this section, linear advection PDE is used to demonstrate the availability of different numerical schemes because of its simple equation structure. The initial conditions and boundary conditions are the same as section 3.2, and the parameters of advection equation are the same as 3.2.2.

3.3.1 Naive forward time center space (FTCS) scheme

From the explicit Euler method, the FTCS scheme has already been mentioned in the section 3.2.2. This scheme has the difference form.

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

The stencil diagram of FTCS scheme shows in figure 14.

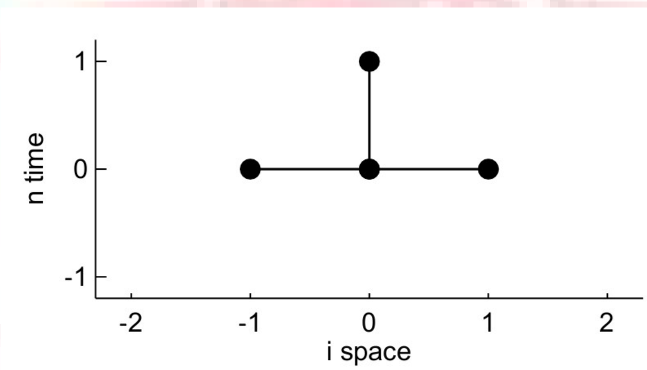

Fig.14 Stencil diagram of FTCS scheme

The simulation result of FTCS scheme is shown in figure 15. Apparently, this method has huge numerical oscillations as we showed in section 3.2.2 that we can't use it to simulate the advection PDE at all.

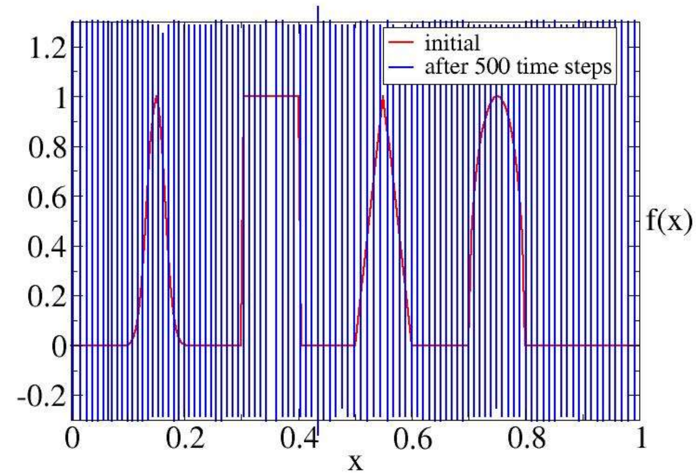

Fig.15 FTCS scheme

3.3.2 Forward time forward space (FTFS) scheme

The FTFS scheme has the difference form.

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

The stencil diagram of FTCS schemes is shown in figure 16.

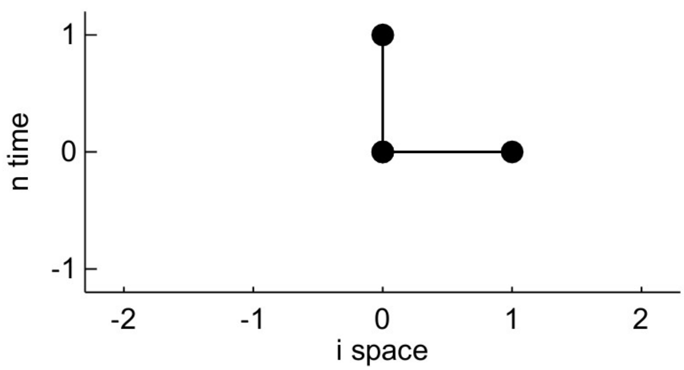

Fig.16 Stencil diagram of FTFS scheme

The simulation result of FTFS scheme is shown in the figure 17. Apparently, this scheme also doesn't work for advection equation, its oscillations grows even faster than FTCS scheme.

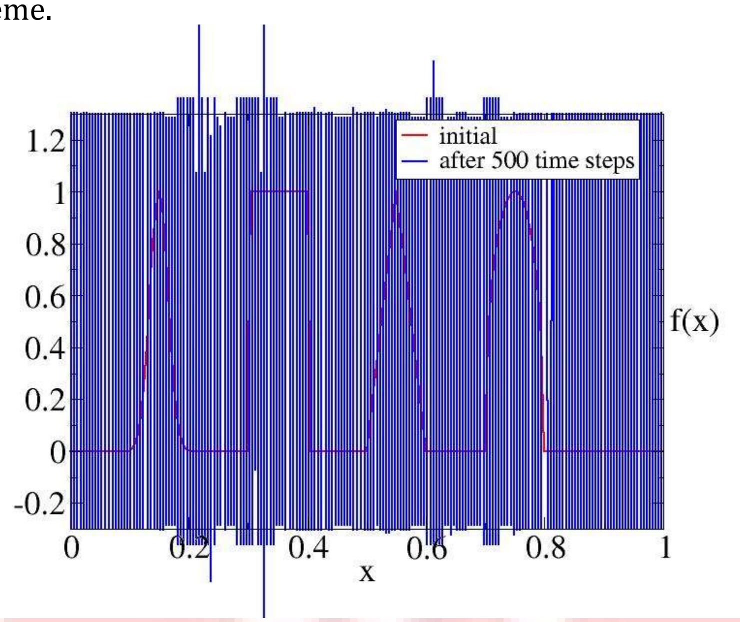

Fig.17 FTFS scheme with v=1

However, if we change its velocity to v = -1, it means that $\frac{v\delta t}{2\delta x} = -0.1$, the result is given in figure 18.

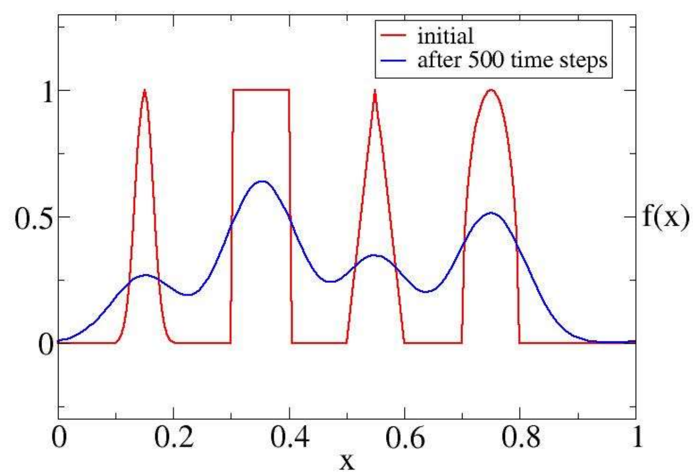

Fig.18 FTFS scheme with v=-1

Figure 18 shows that the FTFS scheme works as the same as FTBS scheme, because these two schemes become identical when we change the velocity of FTFS scheme to negative.

3.3.3 Forward time backward space (FTBS) scheme

The FTBS scheme is also called donor cell scheme. It has the difference form.

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

The stencil diagram of FTBS scheme is shown in figure 19

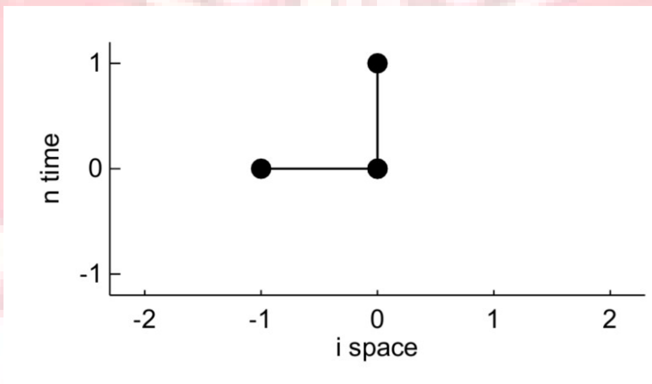

Fig.19 Stencil diagram of FTBS scheme

The simulation result of FTBS scheme is shown in figure 20. This method has better performance than FTCS and FTFS scheme. But this scheme still doesn’t work well enough after long time. This scheme can only ensure converged results. As a result, we need smaller space grids to improve its accuracy.

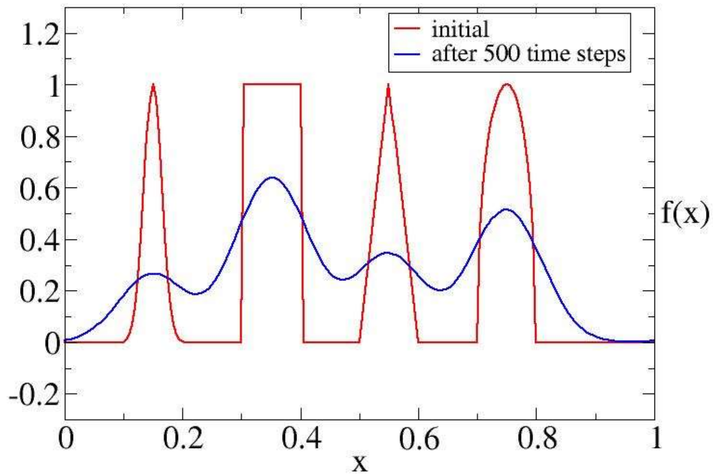

Fig.20 FTBS scheme

3.3.4 Lax-Friedrichs scheme

This scheme has more complicated structure and derivation. It has the difference form.

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

The stencil diagram of Lax-Friedrichs scheme is shown in figure 21.

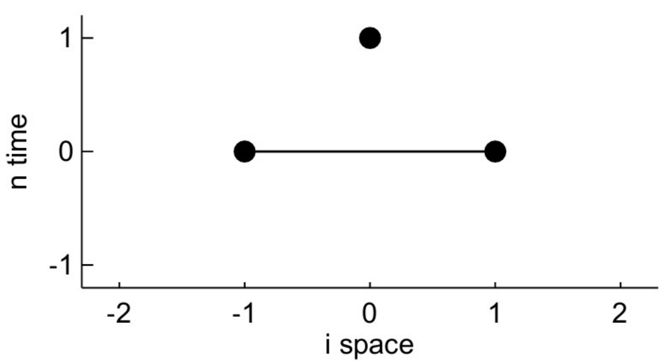

Fig.21 Stencil diagram of Lax-Friedrichs scheme

The simulation result of Lax-Friedrichs scheme is shown in figure 22. There are huge dissipation and zigzag noises when using this scheme.

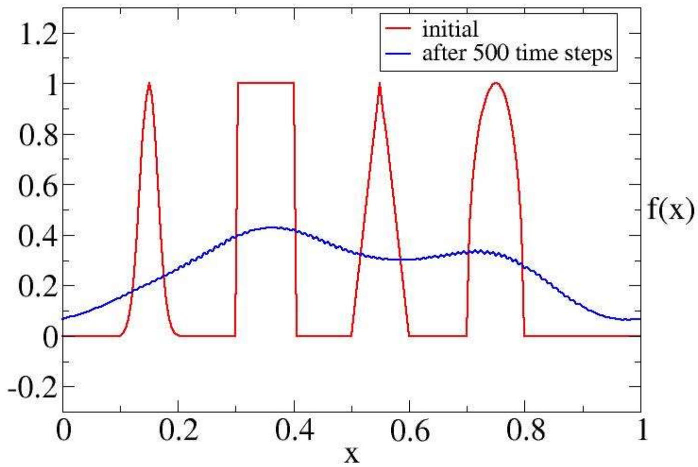

Fig.22 Lax-Friedrichs scheme

3.3.5 Lax-Wendroff scheme

This scheme is developed from Lax-Friedrichs scheme. This scheme has the difference form.

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

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

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

The stencil diagram of Lax-Wendroff scheme shows in figure 23.

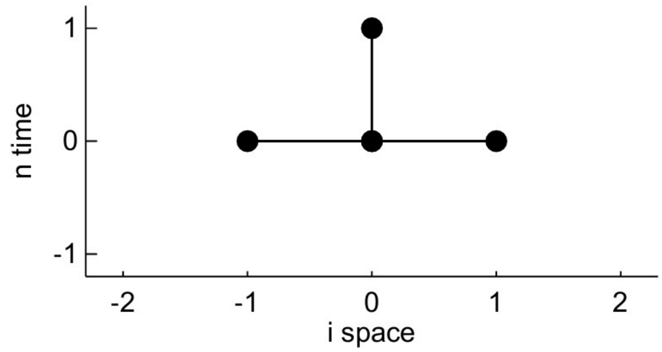

Fig.23 Stencil diagram of Lax-Wendroff scheme

The simulation result of Lax-Wendroff scheme is shown in the figure 24. It shows a smoother solution with overshoots which doesn't grow with time.

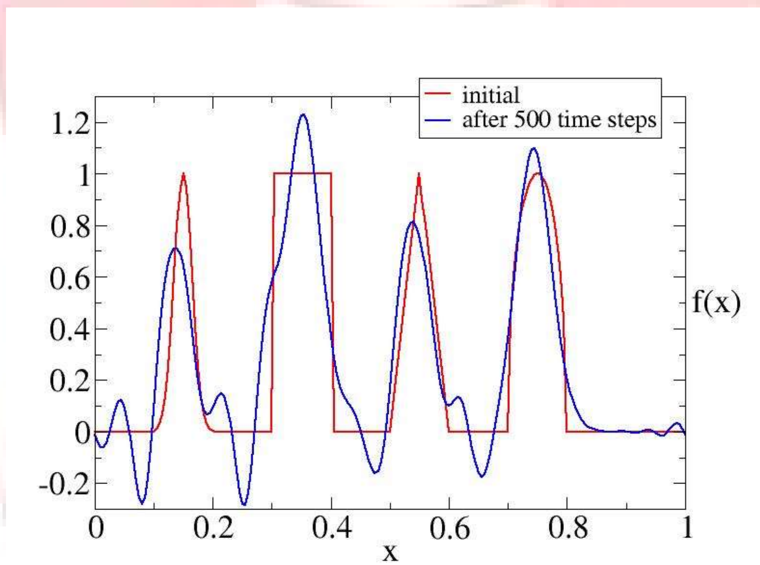

Fig.24 Lax-Wendroff scheme

Figure 25 shows the simulation result of Lax-Wendroff scheme after 20 time steps.

It shows that the overshoots here already happened at the beginning.

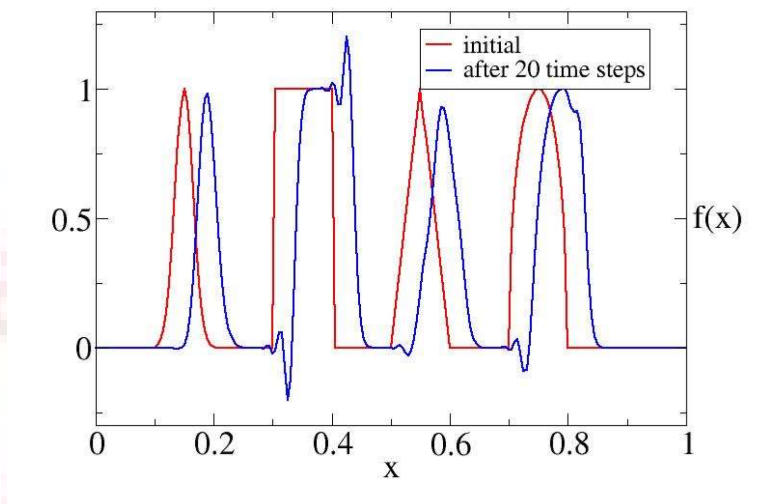

Fig.25 Lax-Wendroff scheme after 20 steps

3.3.6 Beam-Warming scheme

Beam-Warming scheme is basically modified from Lax-Wendroff scheme. This scheme has the difference form.

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

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

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

The stencil diagram of Beam-Warming scheme is shown in figure 26.

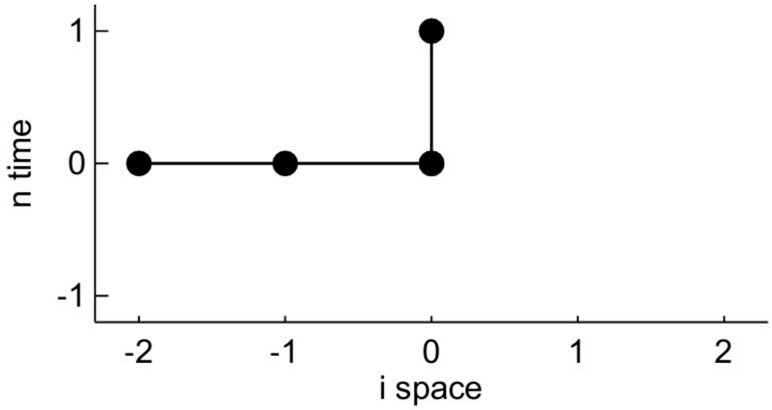

Fig.26 Stencil diagram of Beam-Warming scheme

The simulation result of Beam-Warming scheme is shown in figure 27. The result is similar to Lax-Wendroff scheme. Both of them have overshoot after propagating, but still have a similar shape with initial condition.

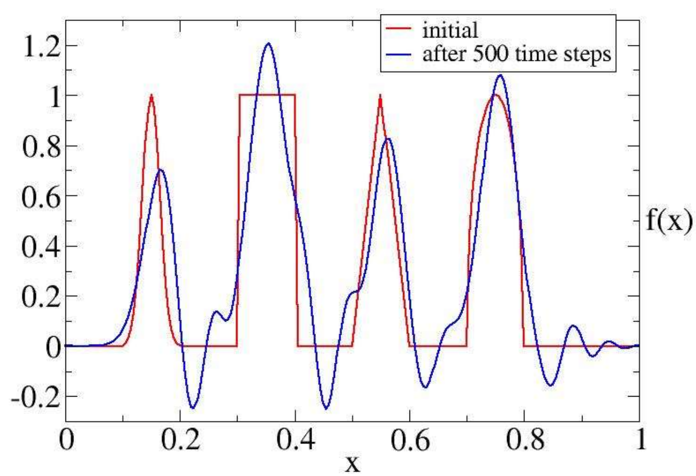

Fig.27 Beam-Warming scheme

Figure 28 shows the simulation result of Beam-Warming scheme after 20 time steps. The shape in this scheme also doesn't change much but with few overshoots.

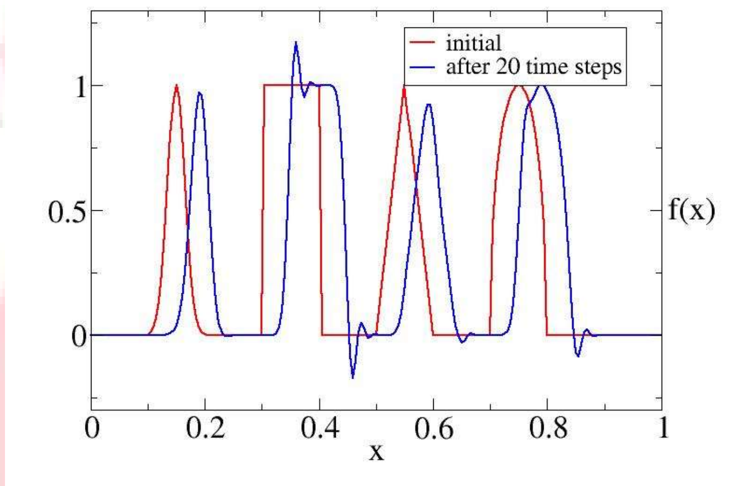

Fig.28 Beam-Warming scheme after 20 steps

3.3.7 Fromm scheme

Fromm scheme combines Beam-Warming scheme and Lax-Wendroff scheme. It has the difference form.

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

$$\boldsymbol{f^n_{i+\frac{1}{2}}} = \boldsymbol{f^n_{i+\frac{1}{2}}}(Lax - Wendroff) + \boldsymbol{f^n_{i+\frac{1}{2}}}(Beam - Warming),$$

$$\boldsymbol{f^n_{i-\frac{1}{2}}} = \boldsymbol{f^n_{i-\frac{1}{2}}}(Lax - Wendroff) + \boldsymbol{f^n_{i-\frac{1}{2}}}(Beam - Warming).$$

The stencil diagram of Fromm scheme is shown in figure 29.

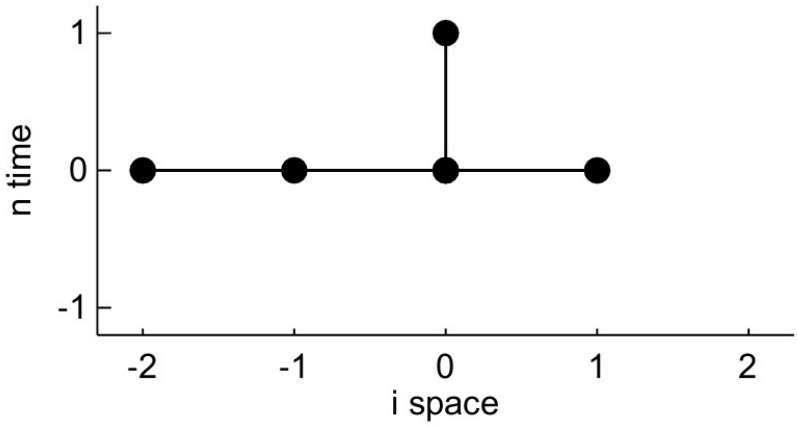

Fig.29 Stencil diagram of Fromm scheme

The simulation result of Fromm scheme is shown in figure 30. This simulation result is the closet to the original initial conditions among all different methods.

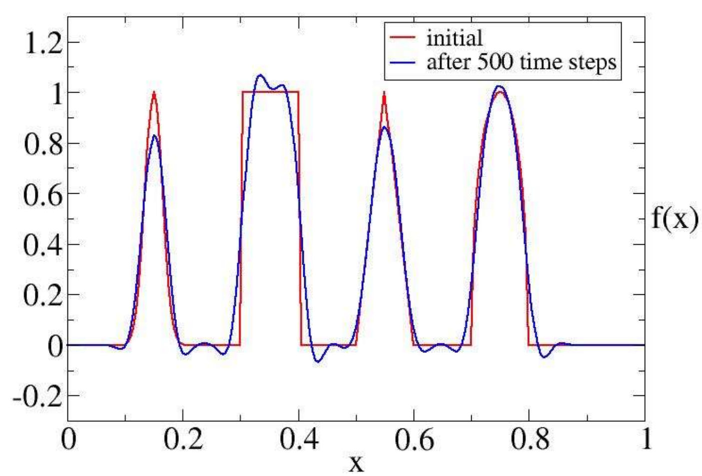

Fig.30 Fromm scheme

3.3.8 Backward time center space (BTCS) scheme

This scheme is just like FTCS scheme but solves the advection equation implicitly. The equation of BTCS scheme has the difference form.

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

The stencil diagram of BTCS scheme is shown in figure 31.

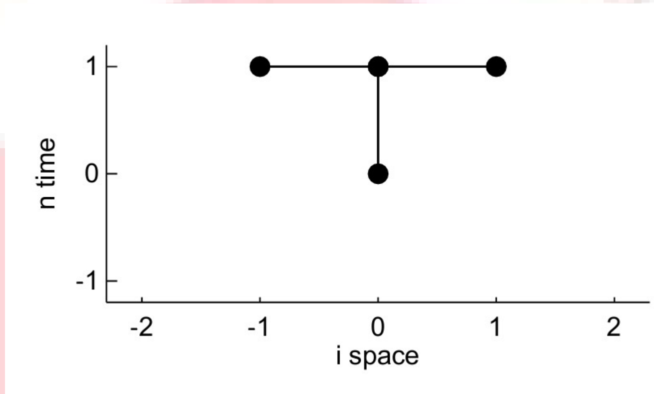

Fig.31 Stencil diagram of BTCS scheme

The simulation result of BTCS scheme is shown in figure 32. This scheme also has a little oscillation and is diffusive heavily. BTCS scheme doesn't work, too.

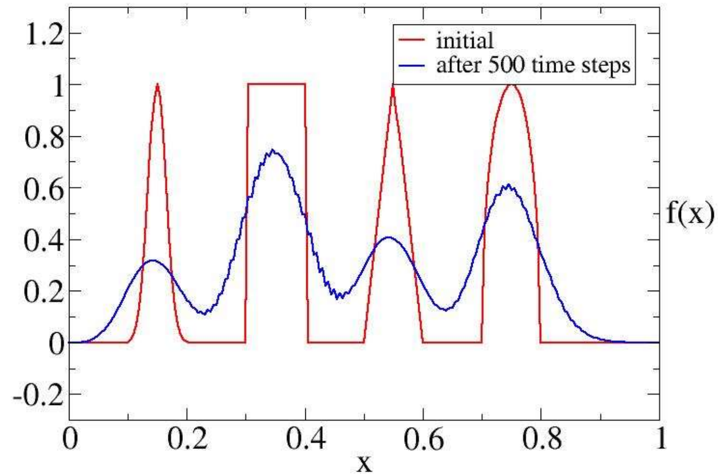

Fig.32 BTCS scheme

Chapter 4 Future work

4.1 Short term goal

In the book "Introduction to Numerical Hydrodynamics", there are some simulation schemes need to be verified, such as piecewise linear method (PLM) scheme to solve nonlinear partial differential equations.

4.2 Long term goal

After practicing lots of numerical schemes and methods to simulate hydrodynamic equations, it's time to simulate the exact physical phenomena in other papers. We will simulate the magnetic and plasma interactions of Solar-terrestrial system in space. We have already found the papers [2][3] related to what we want to simulate. Therefore, after finishing the works now, that is the next goal.

Chapter 5 Summary

After practicing lots of numerical schemes and methods to solve ODE and PDE in the textbook, "Introduction to Numerical Hydrodynamics", is time to go to next step. We need different numerical schemes or even discretization methods to be used in different conditions. To solve the linear advection equation, the best scheme should is Fromm scheme which combines Lax-Wendroff scheme and Beam- Warming scheme. However, my simulation results still have a little different from the results compared with the textbook. The simulation results are possible to solve advection equation are only Fromm scheme, Lax-Wendroff scheme, and Beam-Warming scheme, the other schemes are all have big numerical oscillations or severe overshoot problems.

Reference

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

[3] Takayuki Umeda, Kentaro Togano, Tatsuki Ogino, Two-dimensional full-electromagnetic Vlasov code with conservative scheme and its application to magnetic reconnection, Comp. Phys. Comm. 180, 365–374(2009)