有关 Burgers Equation 的数值求解

Simulations of Burgers Equation

少年班学院
陈 锴 沣  PB09203190
Thomas Gasenzer 教授
二〇一三年六月
中国科学技术大学
University of Science and Technology of China
本科毕业论文
A Dissertation for the Bachelor’s Degree

有关 Burgers Equation 的数值求解
Simulations of Burgers Equation

姓 名 陈 锴 沣
B.S. Candidate Kaifeng Chen
导 师 Thomas Gasenzer 教授
Supervisor Prof.Thomas Gasenzer

二〇一三年六月
June,2013
学士学位论文

中国科学技术大学

有关 Burgers Equation 的数值求解

院 系
少年班学院

姓 名
陈 锵 沣

学 号
PB09203190

导 师
Thomas Gasenzer 教授

二〇一三年六月
University of Science and Technology of China

A Dissertation for the Bachelor’s Degree

Simulations of Burgers Equation

B.S. Candidate  Kaifeng Chen
Supervisor  Prof. Thomas Gasenzer

Hefei, Anhui 230026, China
June, 2013
This work is done in Universität Heidelberg in professor Thomas Gasenzer’s group. I’d like to give my sincere thanks to professor Thomas Gasenzer and Ph.D students Steven Mathey and Markus Karl in helping me with this project. And also I would like to thank professor Matthias Weidemüller, the organizer of the exchange program, for providing me this wonderful chance.

Special thanks to University of Science and Technology of China (USTC) and Center for Quantum Dynamics (CQD) for the financial support.
# 目录

致谢 i

摘要 v

Abstract vii

第一章 Introduction 1

第一节 Background ........................................ 1
第二节 Basic Properties .................................... 2
第三节 Basic Tools .......................................... 3
第四节 Applications ......................................... 4

一、 The Burgers equation in cosmology ......................... 4

二、 The Burgers equation in condensed matter and statistical physics ........................................ 5

三、 The Burgers equation as testing ground for Navier-Stokes ........................................ 6

第二章 1-D Simulation 7

第一节 Numerical Scheme ................................... 7

一、 Basics .................................................... 7

二、 Adams-Bashforth Methods ................................ 9

三、 Package FFTW ............................................ 9

第二节 Rescaling ............................................. 10

第三节 Numerical Results ................................... 12

一、 Example 1: Heat Equation ................................ 12

二、 Example 2: First Shock .................................. 13

三、 Example 3: Decay of Energy ............................ 14

四、 Example 4: Merging of Multiple Shocks ..................... 16
第三章 2-D Simulation

第一节 Numerical Scheme............................................. 21
一、 Basics.......................................................... 21
二、 Adams-Bashforth Methods..................................... 22
三、 Package FFTW.................................................... 23

第二节 Rescaling....................................................... 24

第三节 Numerical Results............................................. 26
一、 Example 1: Comparison With 1D Simulation................. 26
二、 Example 2: A Simple Shock..................................... 28
三、 Example 3: Heat Equation...................................... 30
四、 Example 4: Case With Initial Potential..................... 32

第四章 Summary..................................................... 37

第一节 A Brief Summary............................................... 37
第二节 Future Work.................................................. 38
第三节 Documents & Source Code................................... 38

参考文献............................................................. 39
摘   要

本文是关于 Burgers Equation 的数值求解，其中求解的维度为一维和二维。Burgers Equation 用于天文、流体力学、分形、表面物理等各个领域，本文着重于数值部分。一维和二维的代码编写主要使用 C++ 语言，用谱方法求解，即用快速傅里叶变换 (FFT) 对非线性方程进行变换求解，求解过程中使用了由 MIT 开发的 FFTW 包。在一维的模拟中计算了 shock 的生成、合并以及能量随时间的衰减，并且求出了相关的衰减系数与理论预期进行了对比，重复了文献上的所有结果。二维的模拟计算了二维 shock 的生成，以及舍去非线性项后与二维热扩散方程的对比，最后计算了当初始速度分布是势函数梯度的时候的系统的散度和旋度。

本工作在德国海德堡大学理论物理研究所完成。

关键字: Burgers Equation, 粘滞系数, 模拟, 湍流, 冲击波, 旋度
Abstract

This article is a thesis about the simulations of Burgers Equation. Burgers Equation has wide applications in cosmology, fluid mechanics, fractal and surface physics. This article is mainly about the numerics of Burgers equation. In the work, 1 dimensional and 2 dimensional simulation results are presented by order. All the code are written in C++ and Burgers Equation is solved by Fourier Spectral Method, namely utilizing Fast Fourier Transform (FFT) to solve the nonlinear equation. During the procedure of solving the Burgers, a package named FFTW is used to undertake all the fourier and inverse fourier transforms. In 1D simulation, the forming of shocks, the merging of multiple shocks and the energy decay are calculated. In 2D simulation, the forming of a two dimensional shock, the comparison with 2D heat equation and the initial case with a potential are solved and thoroughly explained.

This work is done at Institute for Theoretical Physics, University Heidelberg, Germany.

Keywords: Burgers Equation, Viscosity, Simulation, Turbulence, Shock, Vorticity
第一章  Introduction

第一节  Background

This article is mainly about the Burgers Equation

\[ \partial_t u + (u \cdot \nabla) u = \nu \nabla^2 u \]  

(1-1)

In the thirties when the Dutch scientist J.M.Burgers introduced the equation in the one-dimensional case\(^1\), he hoped to contribute to the study of turbulence with a simple model which, obviously, has a lot in common with the Navier–Stokes equation\(^2\):

- same type of advective nonlinearity.
- presence of a diffusion term from which a Reynolds number may be defined.
- many invariance and conservation laws in common: invariance under translations in space and time, parity invariance, conservation of momentum and energy (only for \(\nu = 0\) and \(d = 1\))\(^3\).

Such hopes appeared to be shattered when, in the fifties, Hopf\(^3\) and Cole\(^4\) discovered – some say rediscovered – that the Burgers equation can actually be integrated explicitly (I will return to this matter later). Indeed, an important property of the Navier–Stokes equation, not shared by the Burgers equation, is the sensitivity to small changes in the initial conditions in the presence of boundaries or driving forces and at sufficiently high Reynolds numbers. Hence, the

\(^1\)d is the dimension
Burgers equation is not a good model for one of the most important aspects of turbulence: the spontaneous arise of randomness by chaotic dynamics. In spite of this there has been a strong renewal of interest in the Burgers equation, starting in the eighties, for a variety of reasons. The Burgers equation, which obviously describes a compressible flow (in one dimension there exist only trivial incompressible flows), has found many applications in nonlinear acoustics and other nonlinear wave problems. Later there will be some context about the applications of Burgers equation.

第二节 Basic Properties

The most fundamental property of Burgers equation is called "shock". With multiple shocks in the system, research can be about the statistics, which leads to the topic of Burgers turbulence. By definition, Burgers turbulence means the study of random solutions to the Burgers equation. This randomness may arise from random initial conditions or because a random force is added to the r.h.s or the both. The following paragraph will explain why there will be shocks driven by Burgers equation and how they form. The following two pictures will give an intuitive understanding of a shock.

The left picture is the initial velocity profile. Imaging the zero line is at the center of the picture, thus for velocities high than zero, it will point to the right direction.
On the contrary, for the velocities below zero, it will point to the left direction. Therefore, in the center of the picture, we will expect a negative gradient with a large absolute value. When the viscosity $\nu$ at r.h.s of Burgers equation goes to zero, the gradient will be infinite. Here, we call the region of the large gradient a "shock".

shock is the basic property of Burgers equation. All the discussions in the article are about the overall aspects of shocks in very small viscosity limit.

第三节  Basic Tools

Rewrite Burgers equation with an external force here

$$\partial_t u + (u \cdot \nabla) u = \nu \nabla^2 u + f$$  \hspace{1cm} (1-2)

$\nu$ is the viscosity. $f$ is the force per volume acting on the liquid or gas divided by its density. If $u = -\nabla \phi$, i.e. the velocity field is irrotational ($\nabla \times u = -\nabla \times \nabla \phi = 0$), and the force field is the gradient of a potential $f = -\nabla F$, then the Burgers equation can be transformed like following:

$$-\nabla \frac{\partial}{\partial t} \phi + \nabla \phi \cdot \nabla (\nabla \phi) = -\nabla (\nu \nabla^2 \phi) - \nabla F$$

Then using:

$$\nabla \phi \cdot \nabla (\nabla \phi) = \frac{1}{2} \nabla |\nabla \phi|^2$$

Given the boundary conditions, the equation becomes:

$$\frac{\partial}{\partial t} \phi - \frac{1}{2} |\nabla \phi|^2 = \nu \nabla^2 \phi + F$$  \hspace{1cm} (1-3)

Finally with the Hopf-Cole transformation $\phi = 2\nu \ln \theta$, the equation can be written as:

$$\frac{\partial}{\partial t} \theta = \nu \nabla^2 \theta + \frac{1}{2\nu} F \theta$$  \hspace{1cm} (1-4)

If there is not force $F = 0$ and no boundaries, the equation can be solved explicitly:

$$\phi(r, t) = 2\nu \ln \left\{ \frac{1}{(4\pi \nu t)^{3/2}} \int_{R^3} \exp \left[ \frac{1}{2\nu} (\phi_0(a)) - \frac{|r-a|^2}{2t} \right] d^3 a \right\}$$  \hspace{1cm} (1-5)
Where $\phi_0(a)$ is the initial potential. In limit of vanishing viscosity ($\nu \to 0$), obtained by steepest descent, one can get the following "maximum representation"

$$\phi(r, t) = \max_a \left( \phi_0(a) - \frac{|r - a|^2}{2t} \right) \quad (1-6)$$

The maximum will be achieved at one or several points where the gradient of the r.h.s vanishes:

$$r = a - t\nabla \phi_0(a) = a + tv_0(a) \quad (1-7)$$

This means that a particle which starts at $a$ will retain its initial velocity $v_0(a)$. If we describe the movement of particle in both Lagrangian and Eulerian coordinates, then along the Lagrangian trajectory, the velocity is conserved, we have

$$v(r, t) = v_0(a)$$

The map $a \to r$ in (6) is called naive Lagrangian Map, the Jacobian of the map is

$$J(a, t) = \det \left( \delta_{ij} - \frac{1}{t} \frac{\partial^2 \phi_0}{\partial a_i \partial a_j} \right) \quad (1-8)$$

As for the Jacobian, it is not necessarily smooth, the first singularity appears at:

$$t_* = \frac{1}{\max_a [\lambda(a)]}$$

Where $\lambda(a)$ is the largest eigenvalue of the Hessian Matrix $\frac{\partial^2 \phi_0}{\partial a_i \partial a_j}$. The $t_*$ itself is the time point is when shocks begin to form (preshock).

### 第四节 Applications

This section is mainly quoted from reference [2].

一、The Burgers equation in cosmology

The Burgers equation has found interesting applications in cosmology, where it is known, in one instance, as the "Zel’dovich approximation" and, in another instance, as the "adhesion model". Here, we shall give a brief introduction to how
the Burgers equation arises in cosmology. Just after the baryon-photon decoupling in the early Universe, there may have been a rarefied medium formed by collisionless dustlike particles without pressure, interacting only via Newtonian gravity. The gravitational potential is then determined from the fluctuations in mass density by a Poisson equation.

As was shown by Gurbatov and Saichev these are precisely the structures obtained if one modifies the Zel’dovich approximation by requiring that particles should not cross but rather adhere. This adhesion model is just the three-dimensional Burgers equation, taken in the limit of vanishing viscosity. Numerical experiments indicate that the adhesion model reproduces quite well the early skeleton of large-scale structures in N-body numerical simulations. Since dark matter is essentially collisionless, it is not clear at the moment what is the physics behind this agreement which seems to require some viscosity-generating mechanism to prevent – or dramatically slow down – particle crossing. Furthermore, the adhesion model cannot cope with many important aspects of gravitational dynamics. For example, in N-body simulations, one frequently observes the collapse of a filament into an isolated node (cluster). As we shall see, there is nothing of this sort in Burgers dynamics.

二、The Burgers equation in condensed matter and statistical physics

The Burgers equation arises in a number of condensed matter and statistical physics problems and even in non-physics problems such as vehicular traffic. A frequently studied problem is the Kardar–Parisi–Zhang or KPZ equation

$$\partial_t \psi = \frac{1}{2} |\Delta \psi|^2 + \nu \nabla^2 \psi + F$$

(1-9)

which appears in studying the motion of an interface under deposition. Here, \(\psi\) is the vertical displacement of the interface as a function of \(d - 1\) horizontal coordinates and of the time. It is immediately checked, by taking the horizontal gradient of (1-9), that one obtains the Burgers equation with an additional forcing term \(f = -\nabla F\). Burgers equation also arises in studying directed polymers, but
with the time variable now interpreted as a space variable in the direction of main extension of the polymers.

三、 The Burgers equation as testing ground for Navier–Stokes

The Burgers equation, because of its known solutions, is frequently used for testing numerical schemes, particularly those intended for compressible flow. If one is mostly interested in turbulence, as is the case for participants of the present School, Burgers equation turns out to be quite useful for testing—and mostly discarding—certain types of theories of turbulence. Since the Burgers equation has the same type of nonlinearity as the Navier–Stokes equation such methods are typically also applicable to the Burgers equation. Hence it is possible to find what they predict for the latter and to compare the results with those obtained by more reliable methods. From this point of view, that is of using the Burgers equation as testing ground, it is desirable to know the answers to questions similar to those generally asked for Navier–Stokes turbulence.
第二章 1-D Simulation

第一节 Numerical Scheme

一、Basics

The method used in the simulation is called Spectral Methods, or Pseudo-Spectral Method. The one dimensional Burgers equation has the mathematical form

\[
\frac{\partial}{\partial t}u(x,t) + u(x,t) \cdot \frac{\partial}{\partial x}u(x,t) = \nu \frac{\partial^2}{\partial^2 x}u(x,t)
\]  (2-1)

Here, \(\nu\) is the viscosity and \(u\) is a function of both \(x\) and \(t\). In this 1-D simulation, \(u(x,t)\) is stored in a two dimensional array \(u[t][x]\), and \(t\) and \(x\) are integers that parameterize time and space. Then Discrete Fourier Transform (DFT) is used to implement of the spatial derivatives.

The DFT and its inverse form (IDFT) are as follows:

\[
U[t][k] = \sum_{m=0}^{N-1} u[t][m]e^{-i\frac{2\pi km}{N}}
\]  (2-2)

\[
u[t][x] = \frac{1}{N} \sum_{m=0}^{N-1} U[t][m]e^{i\frac{2\pi mx}{N}}
\]

Where \(N\) is the total number of grid points and \(U[t][k]\) is the Fourier transform of \(u[t][x]\).
The first order derivative of \( u[t][x] \) should be like (see (2-3)):

\[
\frac{\partial}{\partial x} u[t][x] = \sum_{0 \leq k < N/2} \frac{2\pi i}{N} k (U[t][k] e^{\frac{2\pi i k x}{N}} - U[t][N - k] e^{-\frac{2\pi i k x}{N}}) \\
:= \sum_{k=0}^{N-1} U'[t][k] e^{\frac{2\pi i k x}{N}}
\]

(2-3)

So the Algorithm_1 to calculate \( \partial u[t][x] / \partial x \) is that:

- firstly calculate \( U[t][k] = DFT\{u[t][x]\} \).
- multiply \( U[t][k] \) by \( \frac{2\pi i}{N} k \) for \( k < N/2 \), by \( \frac{2\pi(k-N)}{N} i \) for \( k > N/2 \), and by 0 for \( k = N/2 \) to obtain \( U'[t][k] \).
- get \( \partial u[t][x] / \partial x = IDFT\{U'[t][k]\} \).

The second order derivative of \( u[t][x] \) is like:

\[
\frac{\partial^2}{\partial^2 x} u[t][x] = -\sum_{0 < k < N/2} \left( \frac{2\pi}{N} k \right)^2 (U[t][k] e^{\frac{2\pi i k x}{N}} + U[t][N - k] e^{-\frac{2\pi i k x}{N}}) - \\
\pi^2 U[t][N/2](-1)^n := \sum_{k=0}^{N-1} U''[t][k] e^{\frac{2\pi i k x}{N}}
\]

(2-4)

So the Algorithm_2 to compute the second derivative is like:

- firstly calculate \( U[t][k] = DFT\{u[t][x]\} \).
- multiply \( U[t][k] \) by \(-\left(\frac{2\pi k}{N}\right)^2\) for \( k \leq N/2 \) and by \(-\left(\frac{2\pi}{N}(k - N)\right)^2\) for \( k > N/2 \) to obtain \( U''[t][k] \).
- get \( \partial^2 u[t][x] / \partial^2 x = IDFT\{U''[t][k]\} \).

In sum, the whole numerical scheme should be like:

- **Step:** 1 getting the \( U[0][k] \).
- **Step:** 2 using Algorithm_1 to get \( \partial u[t][x] / \partial x \).
- **Step:** 3 calculating \( u[t][x] \cdot \partial u[t][x] / \partial x \).
• **Step:** 4 using Algorithm_2 to get $\partial^2 u[t][x]/\partial x$.

• **Step:** 5 solving the Burgers equation in real space by Adams-Bashforth method.

Adams-Bashforth method will be explained in the next section.

二、Adams-Bashforth Methods

In order to get higher accuracy, third order Adams-Bashforth method is used to solve the equation:

$$\frac{\partial u}{\partial t} = f(t, u) \tag{2-5}$$

Where in this case:

$$f(t, u) = -u \frac{\partial}{\partial x} u + \nu \frac{\partial^2}{\partial x^2} u \tag{2-6}$$

Third order Adams-Bashforth method is like this:

$$y_{n+3} = y_{n+2} + h \left( \frac{23}{12} f(t_{n+2}, y_{n+2}) - \frac{4}{3} f(t_{n+1}, y_{n+1}) + \frac{5}{12} f(t_n, y_n) \right) \tag{2-7}$$

When $t = 1, 2$, this is not applicable because there is no such $f(t_{n+1}, y_{n+1}); f(t_n, y_n)$.

So for $t = 1$, regular Euler method is used:

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{2-8}$$

And for $t = 2$, second order Adams-Bashforth method is used:

$$y_{n+2} = y_{n+1} + h \left( \frac{3}{2} f(t_{n+1}, y_{n+1}) - \frac{1}{2} f(t_n, y_n) \right) \tag{2-9}$$

三、Package FFTW

To make the program faster, package **FFTW** is used to calculate the DFT and also the IDFT. **FFTW** is a C library developed by Matteo Frigo and Steven G. Johnson from MIT. Since $u[t][x]$ should be real numbers and $U[t][k]$ may be complex numbers, the following two functions are used:

- *fftw_plan_dft_r2c_1d(int n, double *in, fftw_complex *out, unsigned flags)*.
• `fftw_plan_dft_c2r_1d(int n, fftw_complex *in, double *out, unsigned flags)`.

This first one is the same as DFT and the second one IDFT. To properly use this package, something should be noticed.

From equation:

\[ U[t][k] = \sum_{m=0}^{N-1} u[t][m] e^{-i \frac{2\pi km}{N}} \]  

(2-10)

One can see:

\[ U[t][N - k] = \sum_{m=0}^{N-1} u[t][m] e^{-i \frac{2\pi (N-k)m}{N}} \]

\[ = \sum_{m=0}^{N-1} u[t][m] e^{i \frac{2\pi km}{N}} \]

\[ = (\sum_{m=0}^{N-1} u^*[t][m] e^{-i \frac{2\pi km}{N}})^* \]

Since \( u[t][x] \) is a real number, \( u^*[t][x] = u[t][x] \), then we get:

\[ U[t][N - k] = (\sum_{m=0}^{N-1} u[t][m] e^{-i \frac{2\pi km}{N}})^* = U^*[t][k] \]  

(2-11)

Thus, actually, \( U[t][k] \) only has \( N/2+1 \) independent values, so in the code, \( U[t][k] \) is set to have the size \( TIME \times (N/2+1) \), while \( u[t][x] \) has the size of \( TIME \times N \). In every place where FFTW is used, only half of \( U[t][k] \) is calculated. This also makes Algorithm_1 and Algorithm_2 simpler.

For more details, one can refer to [6] and [7].

第二章  Rescaling

To better analyze the results, we use dimensionless units where the initial energy density and the system size are 1. In this case, we have to rescale all the parameters include \( x, u, t, \nu \) to a new coordinate with \( x', u', t', \nu' \). Before the
calculation, let’s assume that the length in \( x \) coordinate is \( L \), and the total initial energy is:

\[
E = \int_0^L u^2(x, 0) dx
\]  

(2-12)

And here we just ignore the coefficient \( \frac{1}{2} \) in the energy. We can write down the relations between coordinate \( x \) and coordinate \( x' \) in the following way:

\[
x' = x/A, u' = u/B, t' = t/C, \nu' = \nu/D
\]  

(2-13)

\( A, B, C, D \) are constants, and we can do the following steps to get them. First, we have:

\[
x'(L) = L/A = 1 \Rightarrow A = L
\]  

(2-14)

From the assumption that the final energy density is 1:

\[
\int_0^1 u'^2(x', 0) dx' = 1
\]

\[
\Rightarrow \int_0^1 \left( \frac{u(\frac{x}{L}, 0)}{B} \right)^2 d\left( \frac{x}{L} \right) = 1
\]

\[
\Rightarrow \int_0^L \frac{u^2(x, 0)}{B^2} dx = 1
\]

\[
\Rightarrow B^2 = \int_0^L u^2(x, 0) dx = E
\]

Then we get \( B = \sqrt{E} \).

As for \( C \) and \( D \), it is obvious in the coordinate of \( x' \), the Burgers Equation should still be the same, namely:

\[
\frac{\partial u'(x', t')}{\partial t'} + u' \frac{\partial u'(x', t')}{\partial x'} = \nu' \frac{\partial^2 u'(x', t')}{\partial x'^2}
\]  

(2-15)

Then from (2-13), we can get:

\[
\frac{\partial (u/B)}{\partial (t/C)} + (u/B) \cdot \frac{\partial (u/B)}{\partial (x/A)} = (\nu/D) \cdot \frac{\partial^2 (u/B)}{\partial (x/A)^2}
\]

\[
\Rightarrow \frac{C}{B} \frac{\partial u}{\partial t} + \frac{A}{B^2} u \frac{\partial u}{\partial x} = \frac{A^2}{DB^2} \nu \frac{\partial^2 u}{\partial x^2}
\]

Comparing (2-15) with Burgers Equation (2-1), we can get:

\[
\frac{C}{B} = \frac{A}{B^2} = \frac{A^2}{DB}
\]  

(2-16)
Then:

\[ C = \frac{A}{B} = \frac{L}{\sqrt{E}} \quad (2-17) \]
\[ D = AB = L\sqrt{E} \quad (2-18) \]

Thus, the equation (52) can be rewritten as:

\[ x' = x \cdot \frac{1}{L} \]
\[ u' = u \cdot \frac{1}{\sqrt{E}} \]
\[ t' = t \cdot \frac{\sqrt{E}}{L} \]
\[ \nu' = \nu \cdot \frac{1}{L\sqrt{E}} \quad (2-19) \]

Then, the corresponding derivatives in the equation can also be rescaled as:

\[ \frac{\partial u'}{\partial x'} = \frac{L}{\sqrt{E}} \cdot \frac{\partial u}{\partial x} \]
\[ \frac{\partial^2 u'}{\partial x'^2} = \frac{L^2}{\sqrt{E}} \cdot \frac{\partial^2 u}{\partial x^2} \quad (2-20) \]

When it is in the code, \( L = N - 1 \).

### 第三节  Numerical Results

All the results presented here are before rescaling.

### 一、 Example 1: Heat Equation

After neglecting the nonlinear term \( u \frac{\partial u}{\partial t} \) in Burgers equation, the equation becomes heat equation:

\[ \frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \quad (2-21) \]

The function of \( u \) will diffuse driven by this equation. To test the code, parameters are set \( N = 2048, \nu = 100, \Delta t = 10^{-3} \). The result is in figure 2.1:
This gives the proof that the code works well in heat equation.

二、Example 2: First Shock

The second example is about one shock. The parameters are set $\nu = 1000, N = 2500, \Delta t = 10^{-5}$. Actually compared to the scale of the simulation, the viscosity of value 1000 is already a small one. The result indicates very apparent shock front and energy decay (see figure 2.2).
三、Example 3: Decay of Energy

This example proves the theories about the energy decay when time is large enough. According to reference[2], after long time simulation, the velocity profile should be like:

![Diagram](image1)

图 2.3 Multiple shocks after long time

The position of the shocks are random. Here we can only consider a very simple case, see figure 2.4.

![Diagram](image2)

图 2.4 One simple case

In this case, the whole momentum of the system is zero. Thus, in infinite time limit, the energy will decay to zero. No we can only consider the left part of the
shock (the part on the right side of the point where \( u = 0 \)). The velocity profile is linear like and has the slope \( 1/t \). The spacial scale we are considering is \( L \). Actually, in this simple case, we can say the shock position is never random and it should be the point where the velocity is zero. Assuming vanishing viscosity, the length \( L \) will not change with time. If at time \( t_0 \), the maximum velocity is \( v_0 \) and the slope is \( 1/t_0 \), then at time \( t \) we can get:

\[
L = v_0 t_0 = vt
\]  

(2-22)

Then we know:

\[
v(x) = \frac{x}{t}
\]  

(2-23)

Then the energy of the left part will be:

\[
E_l = \frac{1}{2} \int_0^L [v(x)]^2 dx
\]

\[
= \frac{1}{2} \int_0^L (x/t)^2 dx
\]

\[
= \frac{1}{6} L^3 \cdot t^{-2}
\]  

(2-24)

Then the whole energy of the system should be two times of the left part, namely:

\[
E = 2E_l = \frac{1}{3} L^3 \cdot t^{-2}
\]  

(2-25)

Or we can write this in another way:

\[
\ln(E) = -2 \ln(t) + \ln(\frac{1}{3} L^3)
\]  

(2-26)

Equation (2-26) is definitely right in vanishing viscosity limit. To test this case, the parameters set in code are \( N = 4000, \nu = 500, \Delta t = 10^{-5} \). The initial velocity profile and the final velocity profile are shown in figure 2.5 (see next page):

After carefully examining the initial velocity, we can say the shock should be at the point \( N = 2000 \). This is to say \( L = 1000 \). So the equation (2-26) will be:

\[
\ln(E) = -2 \ln(t) + \ln(\frac{1}{3} L^3) = -2 \ln(t) + 19.62
\]  

(2-27)

Then the plot of energy with respect to time is shown in figure 2.6. The linear fit shows the simulation results is:

\[
\ln(E) = -1.83 \ln(t) + 18.689
\]  

(2-28)
Equation (2-28) agrees well with (2-27). The slight differences are because in real simulation the viscosity is not zero, thus the integral (2-24) is not exactly right. But when viscosity is very small, equation (2-26) should be a good approximation.

四、 Example 4: Merging of Multiple Shocks

In the final stage of a system driven by Burgers equation, firstly shocks will appear due to the randomness or fluctuations of the system, then all the shocks will begin to move and merge. This process is unique in Burgers simulation and will reveal some properties of statistics we are interested in.
This example shows the merging of multiple shocks. In the simulation scheme, three shocks are generated and after a long time only one shock remains. The parameters are $N = 8000$, $\nu = 500$, $\Delta t = 10^{-5}$. The initial velocity is like figure 2.7:

![Velocity Profile](image)

图 2.7 $t=0$

After nearly 30s, the three shock will become only one shock. The initial and the final velocity profile are shown in figure 2.8:

![Velocity Profile in two times](image)

图 2.8 $t=0$ & $t=29.85$

To better clarify the whole process, here are four pictures showing the steps of the merging of these three shocks.
Also a movie is generated to show the whole process, see last chapter: Summary. To better evaluate the result, the energy decay is plotted and linear fitted, see figure 2.9:

![Figure 2.9 ln(E) vs ln(t)](image)

The energy in the long time limit shows the decay as $E \sim t^{-1.82}$, which is very
close to the ideal relation in vanishing viscosity limit $E \sim t^{-2}$. Furthermore, the energy decay can be divided into three stages. These stages should be:

- The forming of the 3 shocks.
- The merging of the 3 shocks.
- The decaying of the last shock.
第三章 2-D Simulation

第一节 Numerical Scheme

一、 Basics

The 2D Burgers Equation is:

\[
\frac{\partial u}{\partial t} = -u \cdot \nabla u + \nu \nabla^2 u \tag{3-1}
\]

When it comes to two dimensions, we can write:

\[
u(x, y), w(x, y)\) \tag{3-2}
\]

Where \(v(x, y)\) is the x projection of \(u\) and \(w(x, y)\) is the y projection of \(u\), then the Burgers Equation is:

\[
\frac{\partial v(x, y)}{\partial t} = -\left[ v \frac{\partial v(x, y)}{\partial x} + w \frac{\partial v(x, y)}{\partial y} \right] + \nu \left[ \frac{\partial^2 v(x, y)}{\partial x^2} + \frac{\partial^2 v(x, y)}{\partial y^2} \right] \tag{3-3}
\]

\[
\frac{\partial w(x, y)}{\partial t} = -\left[ v \frac{\partial w(x, y)}{\partial x} + w \frac{\partial w(x, y)}{\partial y} \right] + \nu \left[ \frac{\partial^2 w(x, y)}{\partial x^2} + \frac{\partial^2 w(x, y)}{\partial y^2} \right] \tag{3-4}
\]

Equation (3-3) and (3-4) are coupled with each other. To solve them, following the steps used in 1-D simulation, first all the first order derivatives \(\partial v/\partial x, \partial v/\partial y, \partial w/\partial x, \partial w/\partial y\), and all the second order derivatives \(\partial^2 v/\partial x^2, \partial^2 v/\partial y^2, \partial w^2/\partial x^2, \partial^2 w/\partial y^2\) should be calculated using spectral method. Then Adams-Bashforth method is to be used to solve equation (3-3) and (3-4). In the simulations, all the spatial arrays should have two dimensions like \(*[x][y]*\).
The two dimensional Fourier transform is:

\[ V[\xi][\eta] = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} v[x][y] e^{-\frac{2\pi i x}{N}} e^{-\frac{2\pi i y}{M}} \] (3-5)

\[ v[x][y] = \frac{1}{N M} \sum_{\xi=0}^{N-1} \sum_{\eta=0}^{M-1} V[\xi][\eta] e^{\frac{2\pi i x}{N}} e^{\frac{2\pi i y}{M}} \]

and

\[ W[\xi][\eta] = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} w[x][y] e^{-\frac{2\pi i x}{N}} e^{-\frac{2\pi i y}{M}} \] (3-6)

\[ w[x][y] = \frac{1}{N M} \sum_{\xi=0}^{N-1} \sum_{\eta=0}^{M-1} W[\xi][\eta] e^{\frac{2\pi i x}{N}} e^{\frac{2\pi i y}{M}} \]

\( N \) is the total number of grid points along \( x \) axis and \( M \) along the in \( y \) axis. So the whole process of solving equation (3-3) and (3-4) is like this:

- **Step 1**: Getting \( V[\xi][\eta] \) and \( W[\xi][\eta] \)
- **Step 2**: Using algorithm_1 like 1-D to get \( \partial v / \partial x, \partial v / \partial y, \partial w / \partial x, \partial w / \partial y \).
- **Step 3**: Using algorithm_2 like 1-D to get \( \partial^2 v / \partial x^2, \partial^2 v / \partial y^2, \partial w / \partial x^2, \partial^2 w / \partial y^2 \).
- **Step 4**: Solving Burgers equation using Adams-Bashforth Method.

### Adams-Bashforth Methods

In two dimension, equation (3-3) and (3-4) can be written like this:

\[ \frac{\partial v}{\partial t} = f_v(v, w, t) \] (3-7)

\[ \frac{\partial w}{\partial t} = f_w(v, w, t) \] (3-8)

Where

\[ f_v(v, w, t) = -(v \frac{\partial v}{\partial x} + w \frac{\partial v}{\partial y}) + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \] (3-9)

\[ f_w(v, w, t) = -(v \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial y}) + \nu \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right) \] (3-10)
Then the third order Adams-Bashforth method is:

\[
\begin{align*}
    v_{n+3} &= v_{n+2} + h \left( \frac{23}{12} f_v(v_{n+2}, w_{n+2}, t_{n+2}) - \frac{4}{3} f_v(v_{n+1}, w_{n+1}, t_{n+1}) + \frac{5}{12} f_v(v_n, w_n, t_n) \right) \\
    w_{n+3} &= w_{n+2} + h \left( \frac{23}{12} f_w(v_{n+2}, w_{n+2}, t_{n+2}) - \frac{4}{3} f_w(v_{n+1}, w_{n+1}, t_{n+1}) + \frac{5}{12} f_w(v_n, w_n, t_n) \right)
\end{align*}
\]

(3-11)

For \( t = 1 \), the Euler method is:

\[
\begin{align*}
    v_{n+1} &= v_n + hf_v(v_n, w_n, t_n) \\
    w_{n+1} &= w_n + hf_w(v_n, w_n, t_n)
\end{align*}
\]

(3-12)

For \( t = 2 \), the second order Adams-Bashforth method is:

\[
\begin{align*}
    v_{n+2} &= v_{n+1} + h \left( \frac{3}{2} f_v(v_{n+1}, w_{n+1}, t_{n+1}) - \frac{1}{2} f_v(v_n, w_n, t_n) \right) \\
    w_{n+2} &= w_{n+1} + h \left( \frac{3}{2} f_w(v_{n+1}, w_{n+1}, t_{n+1}) - \frac{1}{2} f_w(v_n, w_n, t_n) \right)
\end{align*}
\]

(3-13)

三、 Package FFTW

In 2D case, two functions in \texttt{FFTW} are used:

- \texttt{fftw_plan_dft_r2c_2d(int n0, int n1, double *in, fftw_complex *out, unsigned flags)}.
- \texttt{fftw_plan_dft_c2r_2d(int n0, int n1, fftw_complex *in, double *out, unsigned flags)}.

By now the flag \texttt{FFTW\_PRESERVE\_INPUT} is not applicable in multi dimensions. To deal with this, temporary arrays are generated to avoid the destroying of input data.

As for the fourier transform, the output array is like \( N \times (M/2+1) \), which means that here we choose \( y \) to be the last dimension. In the mean time, the input real arrays are of dimension \( N \times M \).
第二节  Rescaling

Like 1D case, here we still have to rescale the system. After rescaling, we set the length of \(x\) axis to be 1, and the total energy density also to be 1. So the map from coordinate \(x - y\) to the coordinate \(x' - y'\) will be:

\[
x' = x/A_x, y' = y/A_y, v' = v/B_x, w' = w/B_y, t' = t/C, \nu' = \nu/D \tag{3-14}
\]

Assuming the length in \(x\) direction and in \(y\) direction are \(L_x\) and \(L_y\) respectively. The total initial energy is:

\[
E = \int_0^{L_x} \int_0^{L_y} (v^2 + w^2) dx dy \tag{3-15}
\]

Since:

\[
x'(L_x) = L_x/A_x = 1
\]

Then \(A_x = L_x\). Besides:

\[
x'(L_y) = \frac{L_x}{L_y}
\]

We have \(A_y = L_x = A_x\). We denote them as \(A\). From the equation (3-3), we can see from the first term in right side: \(B_x = B_y\). Let’s denote them as \(B\).

Since the rescaling initial energy density is 1, we have:

\[
\frac{1}{L_y/L_x} \int_0^1 \int_0^{L_y/L_x} (v'^2 + w'^2) dx' dy' = 1
\]

\[
\Rightarrow \frac{L_x}{L_y} \int_0^{L_x} \int_0^{L_y} \left( \frac{v^2}{B_x^2} + \frac{w^2}{B_y^2} \right) dx dy = 1
\]

\[
\Rightarrow B^2 = \frac{L_x}{L_y} \int_0^{L_x} \int_0^{L_y} (v^2 + w^2) dx dy = \frac{L_x}{L_y} E
\]

\[
\Rightarrow B = \sqrt{\frac{L_x}{L_y} E}
\]

Then from equation (3-3), we can see:

\[
\frac{\partial v'(x', y')}{\partial t'} = -\left[ v' \frac{\partial v'(x', y')}{\partial x'} + w' \frac{\partial v'(x', y')}{\partial y'} \right] + \nu' \left[ \frac{\partial^2 v'(x', y')}{\partial x'^2} + \frac{\partial^2 v'(x', y')}{\partial y'^2} \right] \tag{3-16}
\]

Putting (3-14) into (3-16), we get:

\[
\frac{C}{B} \frac{\partial v(x,y)}{\partial t} = - \left[ \frac{A}{B^2} \left( v \frac{\partial v(x,y)}{\partial x} + w \frac{\partial v(x,y)}{\partial y} \right) + \frac{A^2}{BD} \nu \left( \frac{\partial^2 v(x,y)}{\partial x^2} + \frac{\partial^2 v(x,y)}{\partial y^2} \right) \right] \tag{3-17}
\]
Like 1D, we still have:

\[
\frac{C}{B} = \frac{A}{B^2} = \frac{A^2}{BD} \tag{3-18}
\]

Thus:

\[
C = \frac{A}{B} = \frac{L_x}{\sqrt{E \cdot L_y / L_x}} \tag{3-19}
\]

\[
D = AB = \sqrt{E \cdot L_y L_x}
\]

To summarize, the rescaling is:

\[
x' = x \cdot \frac{1}{L_x}
\]

\[
y' = y \cdot \frac{1}{L_x}
\]

\[
v' = v \cdot \frac{1}{\sqrt{E \cdot L_y / L_x}} \tag{3-20}
\]

\[
w' = w \cdot \frac{1}{\sqrt{E \cdot L_y / L_x}}
\]

\[
t' = t \cdot \frac{L_x}{\sqrt{E \cdot L_y / L_x}}
\]

\[
\nu' = \nu \cdot \frac{1}{\sqrt{E \cdot L_y L_x}}
\]

Then the rescaling of all the first order derivatives will be:

\[
\frac{\partial v'}{\partial x'} = \frac{L_x}{\sqrt{E \cdot L_y / L_x}} \frac{\partial v}{\partial x}
\]

\[
\frac{\partial v'}{\partial y'} = \frac{L_x}{\sqrt{E \cdot L_y / L_x}} \frac{\partial v}{\partial y}
\]

\[
\frac{\partial w'}{\partial x'} = \frac{L_x}{\sqrt{E \cdot L_y / L_x}} \frac{\partial w}{\partial x}
\]

\[
\frac{\partial w'}{\partial y'} = \frac{L_x}{\sqrt{E \cdot L_y / L_x}} \frac{\partial w}{\partial y} \tag{3-21}
\]
The rescaling of second order derivatives will be:

\[
\begin{align*}
\frac{\partial^2 v'}{\partial x'^2} &= \frac{L_x^2}{\sqrt{E \cdot L_y/L_x}} \frac{\partial^2 v}{\partial x^2} \\
\frac{\partial^2 v'}{\partial y'^2} &= \frac{L_y^2}{\sqrt{E \cdot L_x/L_y}} \frac{\partial^2 v}{\partial y^2} \\
\frac{\partial^2 w'}{\partial x'^2} &= \frac{L_x^2}{\sqrt{E \cdot L_y/L_x}} \frac{\partial^2 w}{\partial x^2} \\
\frac{\partial^2 w'}{\partial y'^2} &= \frac{L_y^2}{\sqrt{E \cdot L_x/L_y}} \frac{\partial^2 w}{\partial y^2}
\end{align*}
\]  

(3-22)

When it is in the code, \(L_x = N_x - 1, L_y = N_y - 1\).

### 第三节 Numerical Results

一、 Example 1: Comparison With 1D Simulation

The first thing to test the code is to get the same result as in 1D. To make a 2D system like 1D, from equation (3-3) and (3-4), we know that here are two ways to achieve this goal:

1. Setting \(w(x, y) = 0, v(x, y) = v(x)\).
2. Setting \(v(x, y) = 0, w(x, y) = w(y)\).

- In senario (1), all the derivatives along y direction go to zero (\(\partial w/\partial x = 0, \partial w/\partial y = 0\)). Then the 2D Burgers equation is reduced to 1D Burgers equation along x direction. This is a way to test the code in this direction, namely testing all the derivatives and Adams-Bashforth methods.

- In senario (2), all the derivatives along x direction go to zero (\(\partial v/\partial x = 0, \partial v/\partial y = 0\)). The Burgers equation also decouples into 1D to test all the derivatives and Adams-Bashforth method in this direction.

In comparison with 1D, we give the same initial conditions and let 2D and 1D run for the same physical time. The following are the results.
By comparing the results from 2D simulation and 1D simulation, we can see that the curve almost overlaps. This means that the code works well with all the derivatives.
二、 Example 2: A Simple Shock

Firstly, rewrite 2D Burgers equation here:

\[
\begin{align*}
\frac{\partial v(x,y)}{\partial t} & = -\left[ v \frac{\partial v(x,y)}{\partial x} + w \frac{\partial v(x,y)}{\partial y} \right] + \nu \left[ \frac{\partial^2 v(x,y)}{\partial x^2} + \frac{\partial^2 v(x,y)}{\partial y^2} \right] \\
\frac{\partial w(x,y)}{\partial t} & = -\left[ v \frac{\partial w(x,y)}{\partial x} + w \frac{\partial w(x,y)}{\partial y} \right] + \nu \left[ \frac{\partial^2 w(x,y)}{\partial x^2} + \frac{\partial^2 w(x,y)}{\partial y^2} \right]
\end{align*}
\tag{3-23/24}
\]

This example is to simply generate a shock. In this scenario, in order to test the code with coupling Burgers equation, the velocity fields \(v, w\) are all functions of \(x, y\). In detail, the grid is a 1000 \(\times\) 1000 one, and the velocity function is like:

\[
|u(x,y)| = 80\sqrt{2} \cdot \exp \left( -\frac{(x-400)^2 + (y-400)^2}{150^2} \right)
\tag{3-25}
\]

Then, we set:

\[
\begin{align*}
v(x,y) & = 80 \cdot \exp \left( -\frac{(x-400)^2 + (y-400)^2}{150^2} \right) \\
w(x,y) & = 80 \cdot \exp \left( -\frac{(x-400)^2 + (y-400)^2}{150^2} \right)
\end{align*}
\tag{3-26/27}
\]

And:

\[
u(x,y) = v(x,y)\hat{x} + w(x,y)\hat{y}
\tag{3-28}
\]

Where \(\hat{x}\) and \(\hat{y}\) are unit vector along \(x\) and \(y\) direction respectively. In this case, it is obvious to expect a shock along the diagonal direction. The other parameters are \(\Delta t = 10^{-5}\), \(\nu = 1000\) before rescaling.

The following picture shows the initial velocity profile\(^1\).

\(^{1}\)All the velocities mentioned here are \(u = \sqrt{v^2 + w^2}\)
图 3.3 Initial Velocity Distribution

The following four pictures are results during the whole simulation:

It is easy to see along the diagonal direction, a shock is forming and also decaying, which confirms our assumptions.
三、Example 3: Heat Equation

This example is to compare the 2D analytical solution to Heat equation with the solution calculated by the code. In order to get heat equation, we can simply set all the first order derivatives $\partial v/\partial x, \partial v/\partial y, \partial w/\partial x, \partial w/\partial y$ to be zero. Then the Burgers equation is decoupled like this:

$$\frac{\partial v}{\partial t} = \nu \cdot \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)$$
$$\frac{\partial w}{\partial t} = \nu \cdot \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right)$$

(3-29)

Thus, the Burgers equation is reduced to two identical 2 dimensional heat equation. Here in simulation we set:

$$v(x, y) = w(x, y) = 80 \exp\left(-\frac{(x - 250)^2 + (y - 250)^2}{50^2}\right)$$

Then the velocity $u = \sqrt{v^2 + w^2}$ is:

$$u(x, y) = 80\sqrt{2} \cdot \exp\left(-\frac{(x - 250)^2 + (y - 250)^2}{50^2}\right)$$

(3-30)

And it also satisfy the equation:

$$\frac{\partial u}{\partial t} = \nu \cdot \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

(3-31)

The following pictures shows the initial velocity profile of $u(x, y)$ and the velocity distribution when $t = 2.0$ in physical time.
The parameters set in the simulations are $\nu = 250, \Delta t = 10^{-5}, N_x = N_y = 500$.

The numerical solution to the 2D heat equation with this kind of Guassian initial condition is:

$$u(x, y, t) = u \frac{a}{a + 2\nu t} \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2(a + 2\nu t)}\right) \quad (3-32)$$

Where:

$$u = 80\sqrt{2}, a = 1250, \nu = 250, x_0 = 250, y_0 = 250$$

The following two pictures (after rescaling) show the comparison between analytical results and numerical results when $t = 2.0$. The left one is from equation (3-32) and the right from is from simulation.

图 3.4 Comparison
To better analyze the difference between the analytical results and the numerical results, figure 3.4 (after rescaling) shows \( u_A(x, y) - u_N(x, y) \) where \( u_A \) is from analytical solution and \( u_N \) from numerical calculation.

四、 Example 4: Case With Initial Potential

Case with initial potential is trivial in 1D. However, in 2D, the potential only exists if the vorticity of the initial condition is zero. This special case has been discussed for quite a long time. This example is to demonstrate some basic properties.

Here, we assume that the initial velocities are the gradient of a potential, which means:

\[
\begin{align*}
\mathbf{u}(x, y, 0) &= -\nabla \phi \\
\text{(3-33)}
\end{align*}
\]

Then the vorticity of the system should be zero:

\[
\nabla \times \mathbf{u}(x, y, 0) = \left( \frac{\partial v}{\partial y} - \frac{\partial w}{\partial x} \right) \hat{k} = -\nabla \times \nabla \phi = 0 \quad \text{(3-34)}
\]

Here we set \( \phi \) like:

\[
\phi = -150^2 \cdot \exp \left( -\frac{(x - 500)^2 + (y - 500)^2}{150^2} \right)
\]

Then the velocities are:

\[
\begin{align*}
v(x, y, 0) &= 2(500 - x) \cdot \exp \left( -\frac{(x - 500)^2 + (y - 500)^2}{150^2} \right) \\
w(x, y, 0) &= 2(500 - y) \cdot \exp \left( -\frac{(x - 500)^2 + (y - 500)^2}{150^2} \right) \quad \text{(3-35)}
\end{align*}
\]
Here is a plot (3.5) for $w(x, y, 0)$ after rescaling:

![Plot 3.5: $w(x, y, 0)$](image1)

**图 3.5 $w(x, y, 0)$**

Here is another picture (3.6) of the initial potential, the arrows indicates the direction of velocities. Thus, since all the velocities point to the center of the square, we can expect a shock and a very small divergence in the center.

![Plot 3.6: Initial Potential](image2)

**图 3.6 Initial Potential**
The parameters set in the simulation are $N_x = N_y = 1000, \nu = 1000, \Delta t = 10^{-5}$.
The rescaled viscosity is $2.51002 \times 10^{-5}$.
To test the non vorticity condition, we plot a picture (3.7) of $\frac{\partial v}{\partial y} - \frac{\partial w}{\partial x}|_{t=79.8403}$.

![Vorticity](image)

**图 3.7 Vorticity**

We can see that the vorticity on each grid point is almost zero, thus the non vorticity condition is always conserved.

Near the region of shocks, the flux $\nabla \cdot \mathbf{u}(x, y)$ should satisfy the following condition:

$$\nabla \cdot \mathbf{u}(x, y) = \frac{\partial v(x, y)}{\partial x} + \frac{\partial w(x, y)}{\partial y} \ll 0 \quad (3-36)$$
In this case, it is obvious that the shock will appear in the center of the square, the following picture (3.8) is for $\nabla \cdot \mathbf{u}|_{t=79.8403}$ from the simulation.

![Divergence](image)

**图 3.8 Divergence**

The minimum value of $\nabla \cdot \mathbf{u}(x, y)$ locates at the center, and there is where the shock forms.
第四章  Summary

第一节  A Brief Summary

The work is concentrated on the simulations of the Burgers equation. The essence of the methods described in the article is to solve Burgers equation numerically and then discuss the shocks in different dimensions. In 1D simulation cases, four examples are presented with respect to different aspects of Burgers equation. The example about the energy decay, which finally gives the exponent to be $-1.83$, agrees well with the analytical results of $-2$. The energy decay is one of the most important properties when Burgers equation is reduced to 1D. The merging of multiple shocks give an intuitive understanding of how several shocks begin to merge into one. This example pictures the final stage after long time simulation. And based on this, researchers then spend long time on the statistics of the shocks in this final stage. Another important aspect is that after this long computational time, the final velocity distribution will forget the initial condition, which means that the differences of initial condition will become quite small. Thus, the self-similarity of the shocks can be discussed both numerically and analytically. Renomalization group is frequently used here.

For 2D simulation, there are also four examples showing the basics of 2D Burgers equation. The comparison with heat equation validates the code itself and the initial case with a potential finally shows the conservation of zero vorticity of the system. These two examples give us the confidence in the code in solving similar problems in the future.
第二节 Future Work

As has been stated in the introduction, there are still some open problems remaining to be solved.

- Case with vorticity. In the 2D simulation in this article, all the discussions are limited on the case with an initial potential. This means there will be no vorticity all through the whole simulation process. The problem will be what if there are vorticities in the initial condition. And besides, to what extent will the vorticity affect the statistics of the shocks.

- Case with external force. In this scenario, the r.h.s will be added a term with an external force driving on the equation. The same problem exist: what will the shocks be like and what about the statistics.

- Case with randomness. In all the examples shown above, there is no randomness in the system. If the initial velocity is a random distribution, like a guassian distribution, the results will be totally different from what I’ve got.

Already with an efficient code, these problems can be solved one by one. The differences will be in what conditions can we get the physics more directly and how long will it take to run a single simulation.

第三节 Documents & Source Code

The source code of solving Burgers Equation is written in C++, using the package FFTW and Adams-Bashforth method. The code can be viewed here: 1-D Simulation and 2-D Simulation. The short movie about 3 shocks merging can be found on YouTube: Merging of multiple shocks: 1-D. Another movie about the forming of a 2D shock can be found A simple 2D shock. All the documents and the movies can be downloaded from my homepage website: www.stanford.edu/~kfchen.
化学文献


